PROBABILISTIC POINT MATCHING

Gustavo Thebit Pfeiffer

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Ricardo Guerra Marroquim
Daniel Ratton Figueiredo

Rio de Janeiro
Setembro de 2015

PROBABILISTIC POINT MATCHING

Gustavo Thebit Pfeiffer

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Ricardo Guerra Marroquim, D.Sc.

Prof. Daniel Ratton Figueiredo, Ph.D.

Prof. Marcelo Bernardes Vieira, D.Sc.

Profª Maria Eulália Vares, Ph.D.

Dr. Gabriele Sicuro, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2015

Pfeiffer, Gustavo Thebit

Probabilistic Point Matching/Gustavo Thebit Pfeiffer. – Rio de Janeiro: UFRJ/COPPE, 2015.

XII, 148 p.: il.; 29, 7cm.

Orientadores: Ricardo Guerra Marroquim
Daniel Ratton Figueiredo

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 146 – 148.

1. feature matching. 2. asymptotic behavior. I. Marroquim, Ricardo Guerra *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

EMPARELHAMENTO PROBABILÍSTICO DE PONTOS

Gustavo Thebit Pfeiffer

Setembro/2015

Orientadores: Ricardo Guerra Marroquim
Daniel Ratton Figueiredo

Programa: Engenharia de Sistemas e Computação

Devido à própria natureza do processo de formação de imagem — no sentido de que imagens são medidas distorcidas, desordenadas e incompletas de um complexo mundo tridimensional — resolver problemas de emparelhamento é um passo necessário para inúmeras aplicações no campo de visão computacional. No entanto, a maior parte da pesquisa relacionada a emparelhamento no campo é focada em desenvolver algoritmos rápidos e heurísticas, dando pouca atenção à essência dos problemas de emparelhamento. Neste trabalho, apresentamos um arcabouço probabilístico que nos permite derivar métodos ótimos para emparelhamento e provar propriedades fundamentais do problema. Adicionalmente, propomos modelos probabilísticos para os descritores de características do tipo Harris/NCC e SIFT dentro do nosso arcabouço e comparamos os métodos obtidos às alternativas existentes.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PROBABILISTIC POINT MATCHING

Gustavo Thebit Pfeiffer

September/2015

Advisors: Ricardo Guerra Marroquim
Daniel Ratton Figueiredo

Department: Systems Engineering and Computer Science

Due to the very nature of the process of image formation — in the sense that images are incomplete, unordered and distorted measurements of a complex 3D world — solving matching problems is necessary to a number of applications in the field of computer vision. Yet, most research related to matching in the field has focused on developing fast algorithms and heuristics, giving little attention to the essential behavior of matching problems. In this work, we present a probabilistic framework that allows us to derive optimal methods for matching and prove fundamental properties of the problem. In addition, we propose models for Harris/NCC and SIFT feature descriptors using our framework and compare the resulting matching methods to existing approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Matching and applications

Matching is an umbrella term that refers to a family of recurring problems in science and engineering, also known as *correspondence* or *assignment* problems. This kind of problem is particularly ubiquitous in the field of computer vision: The very nature of the process of image formation — in the sense that images are incomplete, distorted and unordered measurements of a complex 3D world — makes *matching* an unavoidable step to a number of applications. As it appears in different forms and with different characteristics to each application, we will refrain from giving a generic, comprehensive definition that encompasses all its variations, and rather exemplify how the problem appears in the different applications.

- In the application of *image stitching* [1], one is given two or more images, taken from the same viewpoint but different angles, and desires to merge them for instance to form a larger image (panorama). In order to overlap images correctly, one needs to, first, know which parts of one image correspond to which parts of the other images, which is a matching problem. This is usually solved with the *feature matching* approach, i.e. finding feature points in images — points that can be easily recognized in the other image — and matching them according to some criteria.

- A similar problem is *uncalibrated stereo*, also known as *structure from motion* [1], in which one is given multiple images of a same object, taken from different viewpoints and possibly different camera models, and pursues a 3D reconstruction of the object. To this end, one needs to know which pixels from each image correspond to the same object point, which is also usually done using feature matching.

- *Calibrated stereo*, also called simply *stereo matching* [1], is easier with respect

to matching than uncalibrated stereo, but it still requires some sort of matching. In this case, as the intrinsic (focal distance, optical center, etc. of each camera) and extrinsic (distance and rotation between cameras) parameters are known, one always knows in which epipolar line a matching point resides. The problem consists in finding the matching point within the line and, using the disparity between the points, perform a 3D reconstruction. Instead of feature matching, other paradigms such as *Markov random fields* are often preferred [2], although methods using feature descriptors have also been proposed [3].

- *Point cloud alignment* is similar to image stitching, but applied to 3D point clouds. In this problem, one is given two or more incomplete point clouds from a real object, e.g. acquired using 3D scanners, and wants to merge them into a complete model. This requires that the point clouds are correctly aligned (i.e. with the correct rotation and translation), which can only be done if one knows which parts of one point cloud correspond to which parts of the other point cloud. When a good initial guess of the alignment is available, iterative *point matching* algorithms[1] such as the *iterative closest point* algorithm [4] may be used, otherwise, feature-based approaches are preferable [5, 6].

- *Tracking multiple points* in a video requires that the moving points are correctly identified and matched in each frame. When matching only two frames, solutions using *minimum bipartite matching* perform well [7]. For multiple frames, however, more sophisticated approaches such as the *k-shortest paths* method may be preferred in order to preserve smooth motion [8, 9].

- Matching may also be used in recognition applications. *Optical character recognition* may be improved if parts of the characters are matched when characters are compared [10], and *fingerprint recognition*, also often requires identifying and matching fingerprint features [11, 12].

While all these applications require matching, their very different characteristics make it inviable to design a framework that generalizes all of them. Rather, we propose a framework that solves a simplified version of the problem, based on a probabilistic model of matching. While this framework does not capture the subtleties of each application, its simplicity enables us to derive optimal algorithms and prove several theoretical properties of the problem.

[1] This sort of algorithm is often called "point set registration" or simply "point matching" in the computer vision literature. Despite the similarity in the name, our framework "probabilistic point matching" has no relation to this class of algorithms: The algorithms we devise in this work cannot be considered "point matching" algorithms in this sense.

Although our motivation were computer vision applications, particularly the *feature matching* approach, our framework is generic enough to be adapted to other areas. The most limiting constraint is that we assume non-matching points behave independently, i.e. the framework does not provide a structure model as *graph matching* approaches (Section 2.3) do.

Our main contributions in this work are:

- A probabilistic framework for matching problems, which we call "*probabilistic point matching*";
- two Bayesian methods — one of polynomial time and another of exponential time — that solve different optimization problems based in this framework;
- analyses on the asymptotic behavior of performance measures of matching methods in our probabilistic framework, with respect to the number of points and the amount of noise;
- the instantiation of our framework in the problem of feature matching, with probabilistic models for Harris/NCC and SIFT feature descriptors; and evaluation comparing to existing methods.

A similar problem to the one we are studying is known in the statistical physics literature as *Euclidean matching* [13, 14], which also studies the asymptotic behavior of a matching problem under a probabilistic model; however, the probabilistic model and measures of interest are different: while we are interested on the ability of the algorithms of producing correct matches, *Euclidean matching* provides no model for match correctness and is mostly concerned with the average matching cost and related properties.

1.2 Structure of this dissertation

This dissertation is divided in three parts.

In Part I, "Models and Algorithms", we present existing algorithms for matching in computer vision, our probabilistic models, and methods based on our framework. In Part II, "Theoretic Results", we analyze the asymptotic behavior of the methods presented in Part I according to the different probabilistic models. Finally in Part III, "Application", we propose models for a computer vision problem and evaluate the resulting method in comparison to existing approaches.

There are also a number of appendices that complement the main text of this dissertation. Particularly Appendix A lists and briefly describes the most frequently used symbols and notations of this dissertation. Appendix F analyzes a related

problem to that of matching, and Appendix E describes an efficient method to deal with the case when the matching cost cannot be computed analytically.

1.3 Remarks on notation

In our probability notation, we do not use the convention of employing capital letters for random variables; the distinction between random and deterministic variables should be inferred by context.

We use capital letters normally to refer to matrices. Also, we do not employ the convention of bold characters for vectors, the distinction between scalar and vector should be inferred by context.

Appendix A may always be referred to in order to recall the definition of a symbol or operator.

Part I

Models and Algorithms

Chapter 2

Matching Strategies

In this chapter, we describe a few matching strategies that are commonly used in computer vision applications. Usually these applications employ heuristic methods that involve finding the nearest neighbor (Sections 2.1.1 and 2.1.3) using some spatial data structure (Section 2.1.4) to speed up the search. However, as matching quality is prioritized instead of computational cost, more powerful solutions may be used (Sections 2.2 and 2.3).

2.1 Greedy approaches

2.1.1 Greedy #1: $O(N^2)$

One of the most simple algorithms used for *feature matching* in computer vision applications works as follows.

We are given two sets of points[1] $P_1, P_2 \subset \mathbb{R}^n$, for usually very high n. For each point $x_1 \in P_1$ we find the most similar point $x_2 \in P_2$ (let us denote this search as $x_2 = \Phi(P_2, x_1) = \arg\min_{x_2' \in P_2} C(x_1, x_2')$, for some cost function $C(x_1, x_2)$, normally the Euclidean distance $||x_1 - x_2||$), and vice versa. A pair of points (x_1, x_2) is added to the match set S if and only if they are the closest to each other (i.e., $\Phi(P_1, \Phi(P_2, x_1)) = x_1$). By analyzing all possible pairs (x_1, x_2), this algorithm costs $O(N^2)$ operations when $|P_1| = |P_2| = N$, or $O(|P_1| \cdot |P_2|)$ in general.

Naturally, many points from both sets will not be added to the match set, so often $|S| < \min\{|P_1|, |P_2|\}$.

[1] What we call a "point" here corresponds to what is usually called in the computer vision literature a *feature descriptor*. It has no relation to the 2D or 3D coordinates of the point; it rather describes characteristics of the point, such as color or gradient histograms of its surroundings.

2.1.2 Greedy #2: $O(N^2 \log N)$

Although not much used in practice, perhaps due to its higher time complexity, this algorithm has the advantage of returning a set match set satisfying $|S| = \min\{|P_1|, |P_2|\}$, while in Greedy #1 $|S| < \min\{|P_1|, |P_2|\}$ was often the case.

This algorithm generates first a set $R = P_1 \times P_2$ and sorts it according to the similarity (i.e. increasingly with the cost function $C(x_1, x_2)$) between the pairs of points. The top pair $(x_1, x_2) = \arg\min_{(x'_1, x'_2) \in R} C(x'_1, x'_2)$ is added to the match set S and all pairs containing x_1 or x_2 are removed from R. This process is repeated until R is empty.

Naturally, because of the greedy nature of the algorithm, the first pairs added to S are very likely to be correct matches, while the latest pairs added will most certainly be false matches. Therefore, stopping somewhere in the middle of the process in order to avoid false matches is not a bad heuristic.

An important property of this method is that all matches produced by Greedy #1 are also necessarily produced by Greedy #2 (i.e., $S_{\text{greedy\#1}} \subseteq S_{\text{greedy\#2}}$). The proof is simple: If $x_2 = \Phi(P_2, x_1)$ and $x_1 = \Phi(P_1, x_2)$, then the pair (x_1, x_2) has a lower cost than any other pair (x_1, x'_2) or (x'_1, x_2). So (x_1, x_2) is the first occurrence of x_1 and x_2 in the sorted set R and therefore the pair is added to $S_{\text{greedy\#2}}$.

Among the pairs that Greedy #2 adds but Greedy #1 does not, there are often both correct and false matches, although we can expect that most of them are false matches. Therefore, although Greedy #2 has a higher *hit count* (number of correct matches), it is expected to have a lower *hit rate* ($\frac{\#\text{correct matches}}{|S|}$) compared to Greedy #1.

2.1.3 Two-nearest neighbors method

The two-nearest neighbors method [15, 16](2-NN) is a popular strategy that uses not only the nearest point $\Phi(P_2, x_1)$, but also the second nearest point $\Phi_2(P_2, x_1) = \arg\min_{x_2 \in P_2 \setminus \Phi(P_2, x_1)} C(x_1, x_2)$ in P_2, where $C(x_1, x_2)$ is normally Euclidean distance. The idea is that, if the nearest point and the second nearest point in P_2 have very similar distances to x_1, then there is a high probability that the nearest point is not the correct match. If the ratio between these distances is very low, then there is a high probability of the match being correct.

So the algorithm adds (x_1, x_2) to the match set S if and only if:

- x_2 is the closest point to x_1 and vice versa, i.e., $x_2 = \Phi(P_2, x_1)$ and $x_1 = \Phi(P_1, x_2)$, and
- the ratio between the distances of the closest point and the second closest point is sufficiently low for both points, i.e., $\frac{||x_1 - \Phi(P_2, x_1)||}{||x_1 - \Phi_2(P_2, x_1)||} < \theta$ and $\frac{||x_2 - \Phi(P_1, x_2)||}{||x_2 - \Phi_2(P_1, x_2)||} < \theta$,

for some threshold $\theta < 1$.

Note that all the matches produced by 2-NN are also necessarily produced by Greedy #1 (the two algorithms are particularly identical when $\theta = 1$).

This property implies that 2-NN has a lower hit count than Greedy #1, although it is expected to have a higher hit rate.

2.1.4 Data structures for matching

Because an $O(N^2)$ cost is prohibitive to many applications, matching is most often done using a greedy algorithm applied on a spatial data structure such as a tree [17] or grid-like [18] data structure. Although this approach can reduce cost to $O(N \log N)$ or $O(C.N)$, search is usually not exact, which may reduce the hit rates.

2.2 Minimum bipartite matching: $O(N^3)$

An approach that makes more effort than the previous methods is to employ minimum bipartite matching, usually solved using the Hungarian algorithm (originally costing $O(N^4)$, later optimized to cost $O(N^3)$, although weakly polynomial solutions faster than $O(N^3)$ have also been proposed [19]).

Supposing the input sets $P_1 = \{X_1^1, X_1^2, X_1^3, ..., X_1^N\}$ and $P_2 = \{X_2^1, X_2^2, X_2^3, ..., X_2^N\}$ have the same sizes, minimum bipartite matching consists of finding a permutation π that solves the following optimization problem:

$$\min_{\pi} \sum_{i=1}^{N} C(X_1^i, X_2^{\pi(i)})$$

for some cost function[2] $C(x_1, x_2)$.

The name "bipartite matching" comes from the interpretation that P_1 and P_2 are two partitions of a bipartite graph, and $C(X_1^i, X_2^j)$ is the weight of the edge that links vertex i of one partition to j of the other. The problem consists then in finding the *matching*, i.e. the set of edges with no common vertices, that minimizes the sum of edge weights. See Figure 2.1 for an illustration.

This combinatorial problem can also be written as a linear programming problem,

[2]Note that differently from the greedy algorithms, here it makes difference whether one chooses Euclidean distance ($||x_1 - x_2||$) or squared Euclidean distance ($||x_1 - x_2||^2$) as cost function.

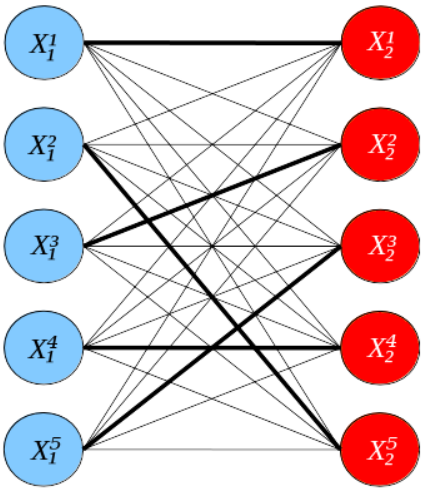


Figure 2.1: Graph interpretation of bipartite matching. Cost is the sum of the matching edges (drawn thicker).

whose variable is a permutation matrix Π:

$$\begin{aligned} \min_{\Pi} \ & \Pi : C \\ \text{subject to: } & \Pi\vec{1} = \vec{1} \\ & \Pi^T\vec{1} = \vec{1} \\ & \Pi_{ij} \geq 0 \end{aligned}$$

where C is a matrix whose entries are $C_{ij} = C(X_1^i, X_2^j)$; ":" denotes the matrix inner product (i.e., $A : B = \sum_{i,j} A_{ij}B_{ij}$), and $\vec{1}$ denotes the vector $\begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$. Although this formulation allows Π to have non-integer entries, the solution will always be a permutation matrix, because the constraint matrix is totally unimodular [20].

The corresponding dual problem is as follows [20]:

$$\begin{aligned} \max_{u,v \in \mathbb{R}^N} \ & u^T\vec{1} + v^T\vec{1} \\ \text{subject to: } & u_i + v_j \leq C_{ij} \end{aligned}$$

The solution of the dual problem is always the same of the primal problem (i.e., the objective functions of their solutions Π and u, v have the same values: $\Pi : C = u^T\vec{1} + v^T\vec{1}$), and $u_i + v_j = C_{ij}$ for the values of i, j where $\Pi_{ij} = 1$.

Instead of directly trying to find the optimal permutation Π, $O(N^3)$ solutions of minimum bipartite matching usually rather work on the dual variables u and v [19], often called the vertex *labelings* of the bipartite graph [21].

An important property of this problem formulation is that multiplying C by a positive constant effects in multiplying $\Pi : C$ by this constant; also, adding a

constant to all members of a row or column in C results in adding the same constant to $\Pi : C$, therefore the optimal permutation Π^* does not change. This means that we can freely replace C by $\tilde{C} = aC + x\vec{1}^T + \vec{1}y^T$, for any $x, y \in \mathbb{R}^N$ (i.e. $\tilde{C}_{ij} = aC_{ij} + x_i + y_j$) without affecting the optimal solution Π^*.

2.3 Graph-based approaches

A more sophisticated approach is to employ graph matching techniques, where instead of matching two sets P_1 and P_2, one wants to match two graphs G_1 and G_2, in such a way that not only matched vertices are similar, but also edges are preserved. It has the advantage of being capable of modeling relations (edges) between input points, but it is also often much costlier than the previous approaches, as formalizations of the problem are in general NP-Hard [22]. It has been used in varied computer vision and pattern recognition applications such as uncalibrated stereo and fingerprint recognition [22].

Chapter 3

Probabilistic Models

In this chapter we describe the basic probabilistic models of our framework. They might not seem very realistic but their simplicity allows us to derive optimal[1] methods, which will be presented in the next chapter, and prove theoretical properties, as will be shown in the subsequent chapters.

3.1 The Direct Model

In our simplest model, we would like to match two given sets P_1 and P_2, both containing N points of $\mathbb{R}^n$, and we assume that points in P_2 are generated by taking a point in P_1 and adding noise.

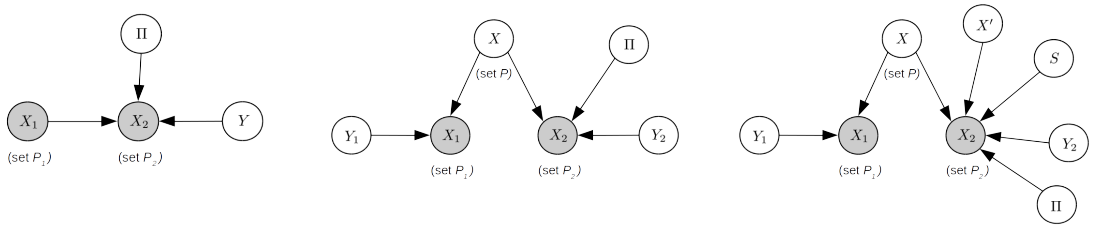
We can represent sets P_1 and P_2 as matrices $X_1, X_2 \in \mathbb{R}^{n\times N}$, so that X_2 is generated from X_1 following

$$X_2 = (X_1 + Y)\Pi$$

where Y is the noise matrix (independent from X_1), and Π is a random permutation matrix in $\mathbb{R}^{N\times N}$ (uniformly distributed in the set of $N \times N$ permutation matrices, i.e. $P[\Pi] = 1/N!$). Writing in this way makes it clear that, if the two sets are represented as two arrays of points, then a priori there is no correlation between the position of a point in one array (i.e. its column index in X_1) and the one of its match (column index in X_2) . Thus, no information is gained by considering the position of the points within their respective arrays; e.g. two points being in the same positions in each array does not make the probability that they match higher.

Additionally, we assume that the columns in X_1 are independent and identically distributed random variables following some distribution with probability density function $p_1(x_1)$. The same applies to the noise (Y) distribution: i.i.d. columns following some probability density function $p_y(y)$.

[1] By optimal, we do not refer to time complexity; we mean that the solutions provided by the methods maximize certain performance criteria.



(a) Direct model

(b) Generator set model

(c) Generator set model with outliers (asymmetric outlier model)

Figure 3.1: Bayesian networks of probabilistic models for matching.

The main disadvantage of this model is that it is asymmetric; i.e., the prior probability distributions $\text{pdf}[x_1]$ of a point $x_1 \in P_1$ and $\text{pdf}[x_2]$ for $x_2 \in P_2$ are different: while $\text{pdf}[x_1] = p_1(x_1)$, $\text{pdf}[x_2] = \{p_1 * p_y\}(x_2)$, where "$*$" denotes convolution in $\mathbb{R}^n$. Particularly, the variance in the distribution of the points in P_2 is higher than the one in P_1, as $\text{Var}[x_2] = \text{Var}[x_1] + \text{Var}[y]$.

Figure 3.1(a) illustrates this model with a Bayesian network.

3.2 Generator Set Model

A more realistic model than the previous one is what we call the *generator set model*. In this model, there is an unknown generator set P, represented by a matrix X, containing i.i.d. points with probability density $\text{pdf}[x] = p(x)$; and the two observed sets P_1 and P_2 are generated independently from P by adding noise, represented respectively by matrices Y_1 and Y_2, also i.i.d. with probability density $p_y(y)$. We can write this in matrix form as follows:

$$X_1 = (X + Y_1)\Pi_1$$

$$X_2 = (X + Y_2)\Pi_2$$

for two random permutation matrices Π_1 and Π_2.

Differently from the previous model, this one is symmetrical, i.e., points in P_1 and P_2 have the same prior probability distributions: $\text{pdf}[x_1] = p_1(x_1)$ for $x_1 \in P_1$, and $\text{pdf}[x_2] = p_2(x_2)$ for $x_2 \in P_2$ where $\forall x : p_1(x) = p_2(x) = \{p * p_y\}(x)$.

In fact, there is some redundancy in this model. It is more reasonable to let $\Pi = \Pi_1^{-1}\Pi_2$ and use instead:

$$X_1 = X + Y_1$$

$$X_2 = (X + Y_2)\Pi$$

See Figure 3.1(b) for the Bayesian network of the resulting model.

3.3 Asymmetric Outlier Model

So far we have not dealt with the possibility of a point not having a match in the other set, which happens frequently in many applications.

To model this we let each point in P_2 have a probability q of being an *outlier*[2]: If the point is an outlier, then we generate again a point with the probability distribution from the points from P and add noise as any other point. This way, the point will be independent from its former match in P_1, and yet have the same prior distribution as the other points in P_2.

In matrix form, this process is written as follows:

$$X_1 = X + Y_1$$

$$X_2 = ((XS + X'(I - S)) + Y_2)\Pi$$

where S is a random diagonal matrix such that $S_{i,i}$ is equal to 0 with probability q and 1 otherwise. Note: S appears twice in the formula above but they refer to the same matrix. Meanwhile, X and X' are independently generated with the same probability distribution. Figure 3.1(c) shows the Bayesian network that underlies this model.

In this model also, points in P_1 and P_2 have the same prior probability distributions.

3.4 Symmetric Outlier Model

The disadvantage of the previous model is that, although points have the same prior distributions, because only the points in P_2 may be generated again as outliers, the distributions of $x_1 \in P_1$ and $x_2 \in P_2$ conditioned on the generator point x from P are different:

$$\begin{aligned} \text{pdf}[x_2|x] &= \text{pdf}[x_2|x, \text{inlier}]P[\text{inlier}] + \text{pdf}[x_2|x, \text{outlier}]P[\text{outlier}] \\ &= (1-q)\text{pdf}[x_2|x, \text{inlier}] + q\text{pdf}[x_2], \end{aligned}$$

while

$$\text{pdf}[x_1|x] = \text{pdf}[x_1|x, \text{inlier}].$$

[2]By "outlier", we mean a point that has no match in the other set. This should not be confused with *false matches*, which are also often called "outliers" in the computer vision literature, particularly in the context of a false match filtering procedure such as RANSAC [23].

So for example, if the distribution of the points in P is a Gaussian distribution with zero mean, and the noise distribution is also Gaussian with zero mean, then x_1 given x follows a Gaussian distribution centered in x, while x_2 given x is a mixture of two Gaussians, one centered in x and the other centered in 0.

One way of correcting this issue is using instead

$$X_1 = XS + X'(I - S) + Y_1$$

$$X_2 = (XS' + X''(I - S') + Y_2)\Pi$$

where X, X' and X'' are independent and identically distributed following the distribution of the points in P, while S and S' are independent and identically distributed with S_{ii} or S'_{ii} having a probability q' of being equal to 0. Note the change in the parameter q to q'.

This model is symmetrical in the sense that the distributions $\text{pdf}[x_1|x]$ and $\text{pdf}[x_2|x]$ are equal. However, it is equivalent to the previous model if we choose q' such that $(1-q')^2 = 1-q$, i.e., in this case, the joint probability density $\text{pdf}[x_1, x_2]$ (given that x_1 and x_2 match according to Π, i.e. $x_1 = X_1^i$ and $x_2 = X_2^j$ for some i, j such that $\Pi_{ij} = 1$) is the same as the one from the previous model. The reason is that, in this model, both x_1 and x_2 are subject to becoming outliers, i.e. with probability $(1-q')^2$ of remaining inliers. In the other model, only x_2 was subject to becoming an outlier, with probability $1-q$ of remaining as inlier. This equivalence means that the same algorithms can be used to solve both models.

3.5 Gaussian noise and properties

The case when noise is Gaussian (i.e. following a multivariate normal distribution) has an important property that will be useful for us later on: In this case, if $x_1 \in P_1$ and $x_2 \in P_2$ are generated from the same point $x \in P$ (and are inliers), then the random variables $M = \frac{x_1+x_2}{2}$ and $D = x_1 - x_2$ are independent, and we can write $\text{pdf}[x_1, x_2] = \text{pdf}[M, D] = \text{pdf}[M]\text{pdf}[D]$. This also means that a means of generating x_1 and x_2 is to generate first M and D independently using the distributions we will derive in this section, and then generate $x_1 = M + D/2$ and $x_2 = M - D/2$.

Let the distribution of the points in P be $\text{pdf}[x] = p(x)$, and the distribution of the noise be $\text{pdf}[y] = p_y(y) = g_\epsilon(y)$, where $g_\epsilon(y) = \frac{e^{-\frac{1}{2}\frac{||y||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}$ denotes the probability density of a Gaussian distribution with zero mean and variance $\epsilon^2 I_{n\times n}$. A point $x_1 \in P_1$ has therefore as prior distribution $\text{pdf}[x_1] = \{p * g_\epsilon\}(x_1)$, where "$*$" denotes convolution in $\mathbb{R}^n$.

We can show that M and D are independent variables as follows:

$$\begin{aligned}
\text{pdf}[M, D] &= \left|\det \begin{bmatrix} \partial x_1/\partial M & \partial x_1/\partial D \\ \partial x_2/\partial M & \partial x_2/\partial D \end{bmatrix}\right| \cdot \text{pdf}[x_1, x_2] \\
&= \left|\det \begin{bmatrix} 1 & 1/2 \\ 1 & -1/2 \end{bmatrix}\right| \cdot \text{pdf}[x_1, x_2] \\
&= \text{pdf}[x_1, x_2] \\
&= \int_{\mathbb{R}^n} \text{pdf}[x]\text{pdf}[x_1|x]\text{pdf}[x_2|x]dx \\
&= \int_{\mathbb{R}^n} p(x) g_\epsilon(x_1 - x) g_\epsilon(x_2 - x) dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} \frac{e^{-\frac{1}{2}\frac{||x_2-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2+||x_2-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n}} dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\cdot\frac{2||x||^2-2\langle x, x_1+x_2\rangle+||x_1||^2+||x_2||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n}} dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\cdot\frac{2||x-\frac{x_1+x_2}{2}||^2-2||\frac{x_1+x_2}{2}||^2+||x_1||^2+||x_2||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n}} dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\cdot\frac{2||x-\frac{x_1+x_2}{2}||^2-\langle x_1, x_2\rangle+\frac{1}{2}||x_1||^2+\frac{1}{2}||x_2||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n}} dx \\
&= \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\left(\frac{||\frac{x_1+x_2}{2}-x||^2}{\epsilon^2/2}+\frac{||x_1-x_2||^2}{2\epsilon^2}\right)}}{(2\pi\epsilon^2)^{n}} dx \\
&= \frac{e^{-\frac{1}{2}\frac{||x_1-x_2||^2}{2\epsilon^2}}}{(2\pi\cdot 2\epsilon^2)^{n/2}} \int_{\mathbb{R}^n} p(x) \frac{e^{-\frac{1}{2}\frac{||\frac{x_1+x_2}{2}-x||^2}{\epsilon^2/2}}}{(2\pi\cdot\epsilon^2/2)^{n/2}} dx \\
&= g_{\sqrt{2}\epsilon}(x_1 - x_2) \int_{\mathbb{R}^n} p(x) \cdot g_{\epsilon/\sqrt{2}}\left(\frac{x_1+x_2}{2} - x\right) dx \\
&= g_{\sqrt{2}\epsilon}(x_1 - x_2) \cdot \left\{p * g_{\epsilon/\sqrt{2}}\right\}\left(\frac{x_1+x_2}{2}\right) \\
&= g_{\sqrt{2}\epsilon}(D) \cdot \left\{p * g_{\epsilon/\sqrt{2}}\right\}(M).
\end{aligned}$$

As their joint probability density function can be decomposed on the product of the probability density functions of each variable, they are shown to be independent, with probability densities of $\text{pdf}[M] = \left\{p * g_{\epsilon/\sqrt{2}}\right\}(M)$ and $\text{pdf}[D] = g_{\sqrt{2}\epsilon}(D)$.

3.5.1 Generalizations

This separation in M and D is also possible in some variations of the Gaussian model.

If noise is anisotropic, i.e. multivariate Gaussian with covariance matrix[3] E^2, then $M = \frac{x_1+x_2}{2}$ and $D = x_1 - x_2$ are independent random variables, with $\text{pdf}[D, M] = \text{pdf}[D]\text{pdf}[M] = \text{pdf}[x_1, x_2]$, $\text{pdf}[D] = g_{\sqrt{2}E}(D)$ and $\text{pdf}[M] = \{p * g_{E/\sqrt{2}}\}(M)$. The derivation is analogous.

If noise is isotropic but asymmetric, i.e. P_1 is generated with a noise of parameter ϵ_1 while P_2 has a noise of ϵ_2, then $D = x_1 - x_2$ and $M = \frac{\epsilon_1^{-2}x_1+\epsilon_2^{-2}x_2}{\epsilon_1^{-2}+\epsilon_2^{-2}}$ are independent variables, with $\text{pdf}[D, M] = \text{pdf}[D]\text{pdf}[M] = \text{pdf}[x_1, x_2]$, $\text{pdf}[D] = g_{\sqrt{\epsilon_1^2+\epsilon_2^2}}(D)$ and $\text{pdf}[M] = \left\{p * g_{\frac{1}{\sqrt{\epsilon_1^{-2}+\epsilon_2^{-2}}}}\right\}(M)$. The derivation also follows the same steps as the symmetric case.

The anisotropic asymmetric case is analogous to the isotropic asymmetric case, but with matrix expressions, i.e. $M = (E_1^{-2} + E_2^{-2})^{-1}(E_1^{-2}x_1 + E_2^{-2}x_2)$, $\text{pdf}[M] = \left\{p * g_{\sqrt{(E_1^{-2}+E_2^{-2})^{-1}}}\right\}(M)$ and $\text{pdf}[D] = g_{\sqrt{E_1^2+E_2^2}}(D)$.[4]

[3]Here E^2 denotes EE^T, and $g_E(x)$ denotes a Gaussian distribution with zero mean and covariance matrix EE^T.

[4]Here, $\sqrt{S}$ for a symmetric positive definite matrix S denotes any matrix M such that $MM^T = S$.

Chapter 4

Bayesian Methods

In this chapter we present two optimization problems that maximize probability metrics on the models described in the previous chapter: The maximum probability problem and the maximum expectation problem.

4.1 The "max-prob" problem

The *maximum probability problem*, or "max-prob" for short, consists of finding a permutation matrix Π that maximizes the posterior probability on the input sets P_1 and P_2. Let us assume there are no outliers for now. We have to solve:

$$\max_{\Pi} P[\Pi|X_1, X_2]$$

Naturally, the prior probability $P[\Pi]$ is equal to $1/N!$ for every Π, but the posterior probability $P[\Pi|X_1, X_2]$ is different, i.e., applying Bayes' law we have:

$$P[\Pi|X_1, X_2] = \frac{\text{pdf}[X_1, X_2|\Pi]P[\Pi]}{\text{pdf}[X_1, X_2]} = \frac{\text{pdf}[X_1, X_2|\Pi]P[\Pi]}{\sum_{\tilde{\Pi}} \text{pdf}[X_1, X_2|\tilde{\Pi}]P[\tilde{\Pi}]}$$

$$= \frac{\text{pdf}[X_1, X_2|\Pi]}{\sum_{\tilde{\Pi}} \text{pdf}[X_1, X_2|\tilde{\Pi}]}$$

Therefore,

$$\arg\max_{\Pi} P[\Pi|X_1, X_2] = \arg\max_{\Pi} \frac{\text{pdf}[X_1, X_2|\Pi]P[\Pi]}{\text{pdf}[X_1, X_2]} = \arg\max_{\Pi} \text{pdf}[X_1, X_2|\Pi].$$

The likelihood $\text{pdf}[X_1, X_2|\Pi]$ is easier to compute, since each pair of points is generated independently. Denoting the i-th columns of X_1 and X_2 as X_1^i and X_2^i, we have:

$$\arg\max_{\Pi} \text{pdf}[X_1, X_2|\Pi] =$$

$$\arg\max_{\Pi} \prod_i \text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi] =$$

$$\arg\min_{\Pi} \sum_i -\log(\text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi]) =$$

$$\arg\min_{\Pi} \Pi : C$$

where $\pi(i) = j \Leftrightarrow \Pi_{ij} = 1$ and C is a cost matrix where $C_{ij} = -\log(\text{pdf}[X_1^i, X_2^j|\Pi_{ij} = 1])$. Therefore, we can solve "max-prob" in $O(N^3)$ operations using the Hungarian algorithm.

4.1.1 Direct model

In the direct model, computing $\text{pdf}[X_1^i, X_2^j|\Pi_{ij} = 1]$ is straightforward, since:

$$\log(\text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi]) =$$

$$\log(\text{pdf}[X_1^i]\text{pdf}[X_2^{\pi(i)}|\Pi, X_1^i]) =$$

$$\log(\text{pdf}[X_1^i]) + \log(\text{pdf}[X_2^{\pi(i)}|\Pi, X_1^i])$$

As shown in Section 2.2, minimum bipartite matching is invariant to adding a constant to every member in a row or column, therefore we can remove the $\log(\text{pdf}[X_1^i])$ term and use only $C_{ij} = -\log(\text{pdf}[X_2^j|\Pi_{ij} = 1, X_1^i]) = -\log(p_y(X_2^j - X_1^i))$. This means that we do not need to know the distribution of the points in P_1 in order to solve "max-prob" in the direct model[1]; we need only the noise distribution.

In the case of Gaussian noise with variance $\epsilon^2 I_{n\times n}$, we obtain

$$C_{ij} = -\log(g_\epsilon(X_2^j - X_1^i)) = \frac{1}{2}\frac{||X_2^j - X_1^i||^2}{\epsilon^2} + \frac{n}{2}\log(2\pi\epsilon^2)$$

Again, because minimum bipartite matching is affine-invariant, we can simply use:

$$C_{ij} = ||X_2^j - X_1^i||^2$$

In other words, solving the direct model with Gaussian noise is the same as solving minimum bipartite matching using squared Euclidean distances as the matching cost.

4.1.2 Generator set model

With the generator set model, we have to compute $\text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi]$, which may not always be tractable, since it requires solving the integral $\int_{\mathbb{R}^n} p(x)p_y(x_1 - x)p_y(x_2 -$

[1]It is not necessary that the points in P_1 are i.i.d., either; the solution to "max-prob" is the same regardless of the distribution in P_1.

$x)dx$.

When noise is Gaussian, however, we can use the distributions of the mean and difference M and D (see Section 3.5.1) giving

$$C_{ij} = -\log\left(\{p * g_{\epsilon/\sqrt{2}}\}\left(\frac{x_1+x_2}{2}\right)\right) - \log(g_{\sqrt{2}\epsilon}(x_1 - x_2)).$$

We still need to compute a convolution, but it may be easy to compute for some distributions. For others, an efficient Monte-Carlo method can be used (See Appendix E). Particularly, if $p(x)$ is a Gaussian distribution with zero mean and $\sigma^2 I$ variance, then we have

$$p * g_{\epsilon/\sqrt{2}} = g_\sigma * g_{\epsilon/\sqrt{2}} = g_{\sqrt{\sigma^2+\epsilon^2/2}}$$

Therefore,

$$C_{ij} = \frac{1}{2}\frac{||\frac{x_1+x_2}{2}||^2}{\sigma^2+\epsilon^2/2} + \frac{1}{2}\frac{||x_1-x_2||^2}{2\epsilon^2} + const.$$

4.1.3 Normalized cost functions

Recall from the invariance properties of minimum bipartite matching (Section 2.2) that we can replace $C(x,y)$ with $\tilde{C}(x,y) = C(x,y)+f(x)+g(x)$, for any two functions f and g, without changing the solution Π^* to the minimum bipartite matching problem. This means that we can replace the cost function with a *normalized* cost function; we will present two normalization methods in the next subsections, which will be useful for us later on.

Normalized Cost Function #1

Let x_1 and x_2 be random variables of sets P_1 and P_2 generated from a same point $x \in P$. Let $h(a,b) = \text{pdf}[x_1 = a, x_2 = b]$. We can replace the original cost function

$$C_{ij} = -\log(h(X_1^i, X_2^j))$$

with the following normalized function:

$$C_{ij} = -\log(h(X_1^i, X_2^j)) + \frac{\log(h(X_1^i, X_1^i)) + \log(h(X_2^j, X_2^j))}{2}$$

$$= -\log\left(\frac{h(X_1^i, X_2^j)}{\sqrt{h(X_1^i, X_1^i)h(X_2^j, X_2^j)}}\right)$$

We will denote this normalized joint probability term later on with the letter H

as:

$$H(x_1, x_2) \triangleq \frac{h(x_1, x_2)}{\sqrt{h(x_1, x_1)h(x_2, x_2)}}$$

This normalized cost function $-\log(H(a, b))$ is interesting mainly for two reasons:

- We can immediately verify that $-\log(H(a, a)) = 0$ for any a.
- As we will show later in Section 5.2.5, $-\log(H(a, b)) \geq 0$ for any a, b. This is a strong result that is valid for any distribution as long as $\text{pdf}[x_1|x] = \text{pdf}[x_2|x]$.

Although both cost functions ($-\log(h(x_1, x_2))$ and $-\log(H(x_1, x_2))$) yield the same results when using the minimum bipartite matching, this is not true if one uses them with a greedy algorithm (i.e. replacing the Euclidean distance with $H(\cdot, \cdot)$). In this case, the properties above make the normalized cost function more attractive to use with greedy algorithms.

Normalized Cost Function #2

Another way of normalizing the cost function is using:

$$C_{ij} = -\log(h(X_1^i, X_2^j)) + \log(p_1(X_1^i)) + \log(p_2(X_2^j))$$

$$= -\log\left(\frac{h(X_1^i, X_2^j)}{p_1(X_1^i)p_2(X_2^j)}\right)$$

We will denote this normalized joint probability term with the letter ζ:

$$\zeta(x_1, x_2) \triangleq \frac{h(x_1, x_2)}{p_1(x_1)p_2(x_2)}$$

One use of this cost function is if one wants to apply "max-prob" to matching two sets of different sizes N_1 and N_2: Suppose without loss of generality that $N_1 < N_2$. A heuristic would be to add dummy points to P_1 so that both sets have the same size. Then, the cost of linking two points would be $-\log h(x_1, x_2)$, while we would have to assign a cost of $-\log(p_2(x_2))$ of linking a dummy point of P_1 to a point $x_2 \in P_2$. However, if we use instead this normalized cost function $-\log \zeta(x_1, x_2)$, we can simply assign cost zero to linking any point x_2 to a dummy point in P_1, which is the same as not adding any dummy point at all (as long as the minimum bipartite matching solver allows sets of different sizes).

4.1.4 Equivalence in Gaussian model

Suppose the points in $x \in P$ follow a Gaussian distribution with zero mean and variance Σ^2 (Σ^2 is an $n \times n$ symmetric positive definite matrix, so that $p(x) = \frac{\exp(-\frac{1}{2}x^T\Sigma^{-2}x)}{(2\pi)^n\sqrt{\det\Sigma^2}}$), and noise is Gaussian with zero mean and a variance of $\epsilon^2 I_{n\times n}$.

In this case we have:

$$h(x_1, x_2) = p_m\left(\frac{x_1 + x_2}{2}\right) g_{\sqrt{2}/\epsilon}(x_1 - x_2)$$

where

$$p_m(M) \triangleq \text{pdf}[M] = \{p * g_{\epsilon/\sqrt{2}}\}(M) = \frac{\exp(-\frac{1}{2}M^T(\Sigma^2 + \epsilon^2 I/2)^{-1}M)}{(2\pi)^n\sqrt{\det(\Sigma^2 + \epsilon^2 I/2)}}$$

and therefore:

$$H(x_1, x_2) = \frac{p_m\left(\frac{x_1+x_2}{2}\right) g_{\sqrt{2}/\epsilon}(x_1 - x_2)}{\sqrt{p_m(x_2)g_{\sqrt{2}/\epsilon}(0)p_m(x_1)g_{\sqrt{2}/\epsilon}(0)}} =$$

$$\frac{\exp\left(-\frac{1}{2}\left(\frac{x_1+x_2}{2}\right)^T(\Sigma^2 + \epsilon^2 I/2)^{-1}\left(\frac{x_1+x_2}{2}\right)\right)\exp\left(-\frac{1}{2}\frac{||x_1-x_2||^2}{2\epsilon^2}\right)}{\sqrt{\exp(-\frac{1}{2}x_1^T(\Sigma^2 + \epsilon^2 I/2)^{-1}x_1)\exp(-\frac{1}{2}x_2^T(\Sigma^2 + \epsilon^2 I/2)^{-1}x_2)}} =$$

$$\exp\left(-\frac{1}{2}\left\|\frac{x_1 + x_2}{2}\right\|^2_{(\Sigma^2+\epsilon^2 I/2)^{-1}} + \frac{1}{4}||x_1||^2_{(\Sigma^2+\epsilon^2 I/2)^{-1}} + \frac{1}{4}||x_2||^2_{(\Sigma^2+\epsilon^2 I/2)^{-1}} - ...\right.$$

$$\left. ... - \frac{1}{2}\frac{||x_1 - x_2||^2}{2\epsilon^2}\right)$$

Using now that[2], for any symmetric positive definite matrix S,

$$\left\|\frac{x_1 + x_2}{2}\right\|^2_S - \frac{1}{2}||x_1||^2_S - \frac{1}{2}||x_2||^2_S =$$

$$\frac{1}{4}||x_1||^2_S + \frac{1}{2}\langle x_1, x_2\rangle_S + \frac{1}{4}||x_2||^2 - \frac{1}{2}||x_1||^2_S - \frac{1}{2}||x_2||^2_S =$$

$$-\frac{1}{4}||x_1||^2_S + \frac{1}{2}\langle x_1, x_2\rangle_S - \frac{1}{4}||x_2||^2 =$$

$$-\frac{1}{4}\left\|x_1 - x_2\right\|^2_S,$$

we obtain:

$$H(x_1, x_2) = \exp\left(\frac{1}{8}\left\|x_1 - x_2\right\|^2_{(\Sigma^2+\epsilon^2 I/2)^{-1}} - \frac{1}{2}\frac{||x_1 - x_2||^2}{2\epsilon^2}\right)$$

$$= \exp\left(-\frac{1}{2}\left\|x_1 - x_2\right\|^2_{\frac{I}{2\epsilon^2} - \frac{(\Sigma^2+\epsilon^2 I/2)^{-1}}{4}}\right).$$

Note that $\frac{I}{2\epsilon^2} - \frac{(\Sigma^2+\epsilon^2 I/2)^{-1}}{4}$ is always positive definite[3].

[2]The notation $||x||^2_S$ refers to $x^T Sx$, while $\langle x, y\rangle_S$ means $x^T Sy$.

[3]This is because if A and B are symmetric positive definite matrices, then $B^{-1} - (A+B)^{-1} = (A+B)^{-1}(AB^{-1}A + A)(A+B)^{-1}$ is also positive definite.

Also remarkably, if the distribution is isotropic, i.e. $\Sigma^2 = \sigma^2 I_{n\times n}$, then the cost function is of the form $-\log(H(x_1, x_2)) = \alpha||x_1 - x_2||^2$ (i.e. $\frac{I}{2\epsilon^2} - \frac{(\Sigma^2+\epsilon^2 I/2)^{-1}}{4} = \left(\frac{1}{2\epsilon^2} - \frac{(\sigma^2+\epsilon^2/2)^{-1}}{4}\right) I = \alpha I$), with $\alpha > 0$.

This means that both the direct model with isotropic Gaussian noise, and the generator set model with isotropic Gaussian distributions (in the generator set and the noise) can be solved using the same method: applying minimum bipartite matching with squared Euclidean distance as cost.

4.1.5 The sorting solution

Another particular case is when the number of dimensions is $n = 1$. In this case, minimum bipartite matching with squared Euclidean distance as cost (therefore the solutions of the direct model with isotropic Gaussian noise and of the generator set model with isotropic Gaussian distributions in P and in the noise) can be solved by sorting the entries of P_1 and P_2 and assigning matches according to their position in the vector (i.e., the i-th member of P_1 after sorting will be assigned to the i-th member in P_2). This means that we can solve "max-prob" in $O(N \log N)$ operations instead of $O(N^3)$ in these cases (see Algorithm 1). Naturally this is only possible with $n = 1$, because in higher dimensions it is not possible to sort points.

Algorithm 1 Minimum bipartite matching with $n = 1$ and squared Euclidean distance as cost:

Sort $P_1 = \{x_1^1, x_1^2, ..., x_1^N\}$ so that $x_1^1 \leq x_1^2 \leq ... \leq x_1^N$;
Sort $P_2 = \{x_2^1, x_2^2, ..., x_2^N\}$ so that $x_2^1 \leq x_2^2 \leq ... \leq x_2^N$;
$S \leftarrow \varnothing$;
for $i = 1, ..., N$ **do**
  Add $s = (x_1^i, x_2^i)$ to S
end for
return S;

The proof is simple. Suppose that "max-prob" has yielded a solution in which the pairs are not ordered, i.e., that there exist $A, B \in P_1$, with $A < B$, and $C, D \in P_2$, with $C < D$, such that A was assigned to D and B to C. Absurd, because assigning A to C and B to D would have lower cost:

$$[(A-D)^2 + (B-C)^2] - [(A-C)^2 + (B-D)^2] =$$
$$[A^2 + B^2 + C^2 + D^2 - 2AD - 2BC] - [A^2 + B^2 + C^2 + D^2 - 2AC - 2BD] =$$
$$2(-AD - BC + AC + BD) =$$
$$2(A-B)(C-D) > 0$$

Notably, the Greedy #2 algorithm (defined in Section 2.1.2) can also be solved in

$O(N \log N)$ when $n = 1$ and Euclidean distance is used as cost; however, it requires a more sophisticated data structure (See Appendix D).

4.2 The "max-expect" problem

Because matching all N pairs correctly (supposing there are no outliers) is usually a too optimistic goal, i.e., usually $\max_\Pi P[\Pi|X_1, X_2] << 1$, we propose to solve instead the *maximum expectation problem* ("max-expect" for short): to maximize the expected *hit count*, i.e. the expected number of correct matches. This is reasonable because the metric we use to compare different methods is usually the hit count, and not the rate of cases in which all pairs were correctly matched (i.e. the most probable permutation is not necessarily the one that has the highest expected hit count).

Supposing the correct permutation is Π, and an algorithm returns a matrix $\tilde{\Pi}$, then the hit count is given by $\Pi : \tilde{\Pi}$, since for every pair (i, j) where $\tilde{\Pi}_{ij} = 1$, it is a correct match if and only if $\Pi_{ij} = 1$. Therefore the optimization problem is written as:

$$\begin{aligned}
&\arg\max_{\tilde{\Pi}} E[\tilde{\Pi} : \Pi|X_1, X_2] \\
&= \arg\max_{\tilde{\Pi}} \sum_{\Pi} \tilde{\Pi} : \Pi P[\Pi|X_1, X_2] \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : \sum_{\Pi} \Pi P[\Pi|X_1, X_2] \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : \sum_{\Pi} \Pi \frac{\text{pdf}[X_1, X_2|\Pi]P[\Pi]}{\text{pdf}[X_1, X_2]} \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : \sum_{\Pi} \Pi \ \text{pdf}[X_1, X_2|\Pi]
\end{aligned}$$

So if we build a cost matrix $\tilde{C} = -\sum_\Pi \Pi \ \text{pdf}[X_1, X_2|\Pi]$, we can solve this using minimum bipartite matching. However, building this cost matrix would cost $O(N!N)$ operations at first glance.

This expression can be further simplified using the permanent of a matrix. The permanent of an $N \times N$ matrix A is by definition:

$$\text{Per}(A) = \sum_{\pi} \prod_{i=1}^{N} A_{i,\pi(i)}$$

where π iterates on all permutations[4] of $\{1, ..., N\}$.

Let R be an $N \times N$ matrix such that $R_{ij} = \text{pdf}[X_1^i, X_2^j|\Pi_{ij} = 1]$, and R_{*ij} be the

[4]We employ lower-case π to denote permutations as functions and upper-case Π as matrices, where $\pi(i) = j \Leftrightarrow \Pi_{ij} = 1$. They refer to the same variable.

$(N-1) \times (N-1)$ matrix obtained by removing the i-th row and the j-th column of R. Then we can write:

$$\begin{aligned}
\tilde{C}_{ij} &= \left(-\sum_{\Pi} \Pi \ \text{pdf}[X_1, X_2|\Pi] \right)_{ij} \\
&= \left(-\sum_{\Pi} \Pi \prod_{k=1}^{n} R_{k,\pi(k)} \right)_{ij} \\
&= -\sum_{\Pi|\Pi_{ij}=1} \prod_{k=1}^{n} R_{k,\pi(k)} \\
&= -R_{ij}\text{Per}(R_{*ij})
\end{aligned}$$

Using this equation, and the fact that the permanent of a matrix can be computed in $O(2^N N)$ operations[5], then we can build $\tilde{C}$ in $O(2^N N^3)$ time. This is a much better time cost than the previous $O(N!N)$, yet still exponential in time.

Interestingly, multiplying a row or column of R by a positive constant results in multiplying $\tilde{C}$ by this same constant, therefore we can replace R by $\tilde{R} = D_1 R D_2$ for arbitrary diagonal (and positive definite) matrices D_1, D_2. This means that we can replace $R_{ij} = \text{pdf}[X_1^i, X_2^j|\Pi_{ij} = 1]$ with a normalized joint probability $R_{ij} = H(X_1^i, X_2^j)$ or $R_{ij} = \zeta(X_1^i, X_2^j)$ (as defined in Section 4.1.3), and obtain the same solution.

This algorithm also provides us directly a confidence measure about a match being correct. The probability that (i, j) is a correct match, given the two sets X_1, X_2, is:

$$P[\Pi_{ij} = 1|X_1, X_2] = \frac{\sum_{\Pi|\Pi_{ij}=1} P[X_1, X_2|\Pi]}{\sum_{\Pi} P[X_1, X_2|\Pi]} = \frac{R_{ij}\text{Per}(R_{*ij})}{\text{Per}(R)} \tag{4.1}$$

Note that $\text{Per}(R)$ can be obtained by summing all the entries of line i or column j of $\tilde{C}$, therefore this confidence measure can be evaluated directly from the cost matrix $\tilde{C}$.

4.3 Case with outliers

The algorithms mentioned above were designed to the case when there are no outliers. Even a small amount of outliers makes their hit rate fall dramatically, so that even using a greedy algorithm is better. We will analyze this phenomenon better in Section 4.5.2.

[5]There is no known polynomial time algorithm to compute a matrix permanent, although approximated (randomized) polynomial solutions do exist [24]. The $O(2^N N)$ algorithm is described in [25].

Recall that using the asymmetric[6] outlier model, points are generated following:

$$X_1 = X + Y_1$$

$$X_2 = ((XS + X'(I - S)) + Y_2)\Pi$$

Ideally, we would like to recover a matrix[7] $\Psi = S\Pi$, which links X_1^i to X_2^j if they were generated from the same point x in the generator set and also are an inlier pair. However, this cannot be reduced to a minimum bipartite matching problem, so we change the approach to try to recover only Π instead. This means that the methods we will describe in this section match pairs of points without discerning if they are inliers or outliers.

In "max-prob", we will solve then $\arg\max_\Pi P[\Pi|X_1, X_2]$. In this case,

$$\text{pdf}[X_1, X_2|\Pi] = \prod_{i=1}^{N} \text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi]$$

$$= \prod_{i=1}^{N} \text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi, S_{ii} = 1]P[S_{ii} = 1] + \text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi, S_{ii} = 0]P[S_{ii} = 0]$$

We know that $P[S_{ii} = 0] = q$. Furthermore, $\text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi, S_{ii} = 1] = h(X_1^i, X_2^j)$ as before and $\text{pdf}[X_1^i, X_2^{\pi(i)}|\Pi, S_{ii} = 0] = \text{pdf}[X_1^i]\text{pdf}[X_2^{\pi(i)}]$, which gives us the following cost function for "max-prob":

$$C_{ij} = -\log(\tilde{h}(X_1^i, X_2^j))$$

or normalized:

$$C_{ij} = -\log(\tilde{H}(X_1^i, X_2^j)) \text{ (normalization method \#1)}$$

$$C_{ij} = -\log(\tilde{\zeta}(X_1^i, X_2^j)) \text{ (normalization method \#2)}$$

where

$$\tilde{h}(X_1^i, X_2^j) = (1 - q)h(X_1^i, X_2^j) + qp_1(X_1^i)p_2(X_2^j);$$

$$\tilde{H}(X_1^i, X_2^j) = \frac{\tilde{h}(X_1^i, X_2^j)}{\sqrt{\tilde{h}(X_1^i, X_1^i)\tilde{h}(X_2^j, X_2^j)}};$$

$$\tilde{\zeta}(X_1^i, X_2^j) = (1 - q)\zeta(X_1^i, X_2^j) + q.$$

[6]Because $\text{pdf}[x_1, x_2]$ is the same distribution for both the asymmetric and symmetric outlier models (using $(1 - q')^2 = 1 - q$), the Bayesian methods for each model are identical, so we derive in this section only the methods for the asymmetric model.

[7]or $\Psi = SS'\Pi$ in the symmetric model

We will show in Section 5.2.5 that, if a generator set model is being used[8], then the normalized cost function #1 with outliers ($-\log(\tilde{H}(x_1, x_2))$) has the same properties of its counterpart in the model without outliers, $-\log(H(x_1, x_2))$: It is equal to zero when $x_1 = x_2$ and is non-negative elsewhere.

As for "max-expect", we will change the objective function to only count inliers:

$$\begin{aligned}
&\arg\max_{\tilde{\Pi}} E[\tilde{\Pi} : S\Pi | X_1, X_2] \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : E[S\Pi | X_1, X_2] \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : \sum_{\Pi} \sum_{S} S\Pi \ \mathrm{pdf}[X_1, X_2 | \Pi, S] P[S] P[\Pi] \\
&= \arg\max_{\tilde{\Pi}} \tilde{\Pi} : \sum_{\Pi} \sum_{S} S\Pi \ \mathrm{pdf}[X_1, X_2 | \Pi, S] P[S]
\end{aligned}$$

So our cost matrix entries will be:

$$\begin{aligned}
\tilde{C}_{ij} &= \left(-\sum_{\Pi} \sum_{S} S\Pi \ \mathrm{pdf}[X_1, X_2 | \Pi, S] P[S] \right)_{ij} \\
&= -\sum_{\Pi | \Pi_{ij}=1} \sum_{S | S_{ii}=1} \mathrm{pdf}[X_1, X_2 | \Pi, S] P[S] \\
&= -P[S_{ii} = 1] \sum_{\Pi | \Pi_{ij}=1} \mathrm{pdf}[X_1, X_2 | \Pi, S_{ii} = 1] \\
&= -P[S_{ii} = 1] \mathrm{pdf}[X_1^i, X_2^j | \Pi_{ij} = 1, S_{ii} = 1] \sum_{\Pi | \Pi_{ij}=1} \prod_{k \neq i} \mathrm{pdf}[X_1^k, X_2^{\pi(k)} | \Pi] \\
&= -(1-q) R_{ij} \mathrm{Per}(((1-q)R + q\bar{R})_{*ij})
\end{aligned}$$

where $R_{ij} = h(X_1^i, X_2^j)$ and $\bar{R}_{ij} = p_1(X_1^i) p_2(X_2^j)$.

As one may see, the time complexities of algorithms "max-prob" and "max-expect" remain $O(N^3)$ and $O(2^N N^3)$ in the case with outliers.

4.3.1 Numerical issues

Because "max-prob" cost functions with outliers take the form $-\log(A + B)$, calculation may be numerically unstable if not carefully implemented. In order to implement these cost functions we use the function

$$\mathrm{lmin}(a, b) \triangleq -\log(e^{-a} + e^{-b}) \tag{4.2}$$

[8]The generator set model applies when Y_1 and Y_2 have the same distributions, and therefore $p_1(x) = p_2(x)$, while the direct model would have $Y_1 = 0$ and $p_1(x) \neq p_2(x)$. Although normalized cost function #1 can also be used with the direct model, it provides no benefits in this case.

which is computed using:

$$\text{lmin}(a, b) = \min\{a, b\} - \log(1 + e^{-|a-b|}).$$

Because the first term above is trivial to compute and second term is always within the range $(0, \log(2)]$, this method is less prone to numerical issues than applying Equation 4.2 directly.

4.3.2 Discerning outliers

While the outlier model versions of "max-prob" and "max-expect" were not designed to detect outliers, once the permutation matrix Π has been found, it is not difficult to infer the outlier selection matrix S: We can model it for instance as:

$$\arg\max_S P[S|X_1, X_2, \Pi] = \arg\max_S \prod_i P[S_{ii}|X_1^i, X_2^{\pi(i)}] =$$

$$\arg\max_S \prod_i \text{pdf}[X_1^i, X_2^{\pi(i)}, S_{ii}]$$

which, using

$$\text{pdf}[X_1^i, X_2^{\pi(i)}, S_{ii} = 1] = (1 - q)h(X_1^i, X_2^{\pi(i)})$$

$$\text{pdf}[X_1^i, X_2^{\pi(i)}, S_{ii} = 0] = qp_1(X_1^i)p_2(X_2^{\pi(i)})$$

has as solution

$$S_{ii} = \begin{cases} 1, & \text{if } (1 - q)h(X_1^i, X_2^{\pi(i)}) > qp_1(X_1^i)p_2(X_2^{\pi(i)}); \\ 0, & \text{otherwise.} \end{cases}$$

or equivalently:

$$S_{ii} = \begin{cases} 1, & \text{if } \frac{h\left(X_1^i, X_2^{\pi(i)}\right)}{p_1\left(X_1^i\right)p_2\left(X_2^{\pi(i)}\right)} > \theta; \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

with $\theta = \frac{q}{1-q}$. In other words, we are thresholding on the normalized cost function #2: $\zeta(x_1, x_2) > \theta$ implies $-\log(\tilde{\zeta}(x_1, x_2)) = -\log(q + (1 - q)\zeta(x_1, x_2)) < -\log(q + (1 - q)\theta) = -\log(2q)$.

Another way of modeling the problem of discerning outliers is to assign different costs to false positives and false negatives. In this case, the classification method is the same (i.e. applying Equation 4.3), but with a different threshold θ, which will further depend on the discrepancy of how one penalizes false positives and false negatives.

4.4 Parameters

Notably, each method, either "max-prob", "max-expect" or the greedy methods, with different probabilistic models require different parameters to perform matching. Table 4.1 summarizes the cost functions and parameters for each method and probabilistic model, in the case of isotropic Gaussian distributions.

4.5 Synthetic experiments

In this section we perform numerical (synthetic) experiments to evaluate the behavior of the proposed methods.

4.5.1 "Max-prob" and "max-expect"

In this experiment, we evaluate the differences between "max-prob" and "max-expect" in terms of expected hit count and probability of hitting all pairs. The purpose is to show that "max-prob" excels at the latter metric, and "max-expect" at the former, although the difference is very small.

We use the direct model with isotropic Gaussian distributions ($p(x) = g_\sigma(x)$ and $p_y(y) = g_\epsilon(y)$) and no outliers. We fixed $n = 2$, and for each $N \in \{3, 5, 7, 9\}$ and $\epsilon/\sigma \in \{.25, .5, 1, 1.5\}$, we generate 10^6 different sets P_1 and P_2 and run "max-prob" and "max-expect" for each pair of sets. The method uses always the same values of σ and ϵ used to generate the sets. We estimate then the average difference in hit count $E[\#\text{hits}_{\text{max-expect}} - \#\text{hits}_{\text{max-prob}}]$ (Table 4.2) and difference in probability of hitting all points $E[\delta_{\#\text{hits}_{\text{max-expect}},N} - \delta_{\#\text{hits}_{\text{max-prob}},N}]$ (Table 4.3), where $\delta_{a,b}$ denotes the Kronecker delta ($\delta_{a,b} = 1$ if $a = b$ or 0 otherwise). In these tables, when we write $x \pm \delta x$, x is an estimator of $E[X]$ for a random variable X, while δx estimates 3 times the standard deviation of the mean estimator, i.e. $3\sqrt{\frac{\text{Var}[X]}{\#\text{samples}}}$ (where $\#\text{samples} = 10^6$). We also compare them to the Greedy #2 algorithm (defined in Section 2.1.2), which uses Euclidean distance as cost.

From these two tables we see that "max-expect" has the best expected hit count of the three algorithms in all cases, although the difference compared to "max-prob" is of about .01% of N. On the other hand, "max-prob" matches all pairs correctly more often than "max-expect", and the difference is also of the order of .01%. Both methods outperform Greedy #2 in both metrics, with discrepancies of much greater magnitude.

Table 4.1: Cost function and required parameters for each method (isotropic Gaussian distributions case)

method	model	outliers	cost function	parameters
Greedy #2	*	*	$\|\|x_1 - x_2\|\|^2$ (by default)	–
"max-prob"	direct	without	$\|\|x_1 - x_2\|\|^2$	–
"max-prob"	generator set	without	$\|\|x_1 - x_2\|\|^2$	–
"max-expect"	direct	without	$-R_{ij}\text{Per}R_{*ij}$, with $R_{ij} = e^{-\frac{\|\|x_1-x_2\|\|^2}{2\epsilon^2}}$	ϵ
"max-expect"	generator set	without	$-R_{ij}\text{Per}R_{*ij}$, with $R_{ij} = e^{-\frac{1}{2}\left(\frac{\|\|x_1-x_2\|\|^2}{2\epsilon^2+\epsilon^4/\sigma^2}\right)}$	$2\epsilon^2 + \epsilon^4/\sigma^2$
"max-prob"	direct	with	$-\log\Big((1-q)g_\epsilon(x_1 - x_2) + qg_{\sqrt{\sigma^2+\epsilon^2}}(x_2)\Big)$	q, σ and ϵ
"max-prob"	generator set	with	$-\log\Big((1-q)g_{\sqrt{2}\epsilon}(x_1 - x_2)g_{\sqrt{\sigma^2+\epsilon^2/2}}(\frac{x_1+x_2}{2})$ $... + qg_{\sqrt{\sigma^2+\epsilon^2}}(x_1)g_{\sqrt{\sigma^2+\epsilon^2}}(x_2)\Big)$	q, σ and ϵ
"max-expect"	direct	with	$-R_{ij}\text{Per}(((1-q)R + q\bar{R})_{*ij})$, with $R_{ij} = g_\epsilon(x_1 - x_2)$ and $\bar{R}_{ij} = g_{\sqrt{\sigma^2+\epsilon^2}}(x_2)$	q, σ and ϵ
"max-expect"	generator set	with	$-R_{ij}\text{Per}(((1-q)R + q\bar{R})_{*ij})$, with $R_{ij} = g_{\sqrt{2}\epsilon}(x_1 - x_2)g_{\sqrt{\sigma^2+\epsilon^2/2}}(\frac{x_1+x_2}{2})$ and $\bar{R}_{ij} = g_{\sqrt{\sigma^2+\epsilon^2}}(x_1)g_{\sqrt{\sigma^2+\epsilon^2}}(x_2)$	q, σ and ϵ

Table 4.2: Average hit count comparison between "max-prob", "max-expect" and Greedy #2 (numerically computed using pseudorandom numbers and 10^6 samples).

N	ϵ/σ	$E[\#\text{hits}_{\text{max-prob}}]$	$E[\#\text{hits}_{\text{max-expect}} - \#\text{hits}_{\text{max-prob}}]$	$E[\#\text{hits}_{\text{greedy\#2}} - \#\text{hits}_{\text{max-prob}}]$
3	0.25	2.91095	0±7.70714e-05	-0.066645±0.00147792
3	0.5	2.69399	2.1e-05±0.000247732	-0.192271±0.00254402
3	1	2.2134	0.000347±0.000565012	-0.322103±0.00351658
3	1.5	1.87822	0.000268±0.000714378	-0.312903±0.0038172
5	0.25	4.70986	9e-05±0.00023098	-0.20362±0.00256711
5	0.5	4.05587	0.000841±0.000670058	-0.483254±0.00395989
5	1	2.86885	0.002799±0.00117229	-0.574369±0.00461069
5	1.5	2.22337	0.002719±0.00125407	-0.463809±0.00458112
7	0.25	6.40292	0.000383±0.000408142	-0.391867±0.00353581
7	0.5	5.15633	0.002994±0.0010777	-0.77287±0.00498171
7	1	3.26422	0.004039±0.00156142	-0.720004±0.00521307
7	1.5	2.41014	0.003497±0.00152705	-0.540693±0.00496357
9	0.25	8.00197	0.000361±0.000609354	-0.617015±0.00439875
9	0.5	6.05293	0.005451±0.00145562	-1.02847±0.00576467
9	1	3.53871	0.007157±0.00182569	-0.815486±0.00561067
9	1.5	2.54075	0.004517±0.00169325	-0.598763±0.00522647

Table 4.3: Comparison of "max-prob", "max-expect" and Greedy #2 methods in terms of the probability of hitting all N points (numerically computed using pseudorandom numbers and 10^6 samples).

N	ϵ/σ	$P[\#\text{hits}_{\text{max-prob}} = N]$	$E[\delta_{\#\text{hits}_{\text{max-expect}},N} - \delta_{\#\text{hits}_{\text{max-prob}},N}]$	$E[\delta_{\#\text{hits}_{\text{greedy\#2}},N} - \delta_{\#\text{hits}_{\text{max-prob}},N}]$
3	0.25	0.955865	-5e-06±3.68646e-05	-0.032539±0.000728687
3	0.5	0.851468	-7.5e-05±0.0001183	-0.089576±0.00122009
3	1	0.634306	-0.000264±0.000266577	-0.137624±0.00161016
3	1.5	0.493419	-0.000518±0.000333226	-0.126004±0.00169251
5	0.25	0.861588	-3.2e-05±0.0001053	-0.089904±0.00118418
5	0.5	0.595566	-0.000341±0.000277024	-0.164881±0.00157367
5	1	0.246824	-0.000436±0.000389467	-0.117716±0.00137949
5	1.5	0.122837	-0.0003±0.000349264	-0.063869±0.0010915
7	0.25	0.733492	-8.1e-05±0.00017304	-0.145162±0.00145733
7	0.5	0.347209	-0.000511±0.000354492	-0.156764±0.00147617
7	1	0.06484	-0.000423±0.0002956	-0.043456±0.000785311
7	1.5	0.018132	-5.1e-05±0.000186652	-0.012968±0.000439291
9	0.25	0.592457	-0.000301±0.000233979	-0.18041±0.00156964
9	0.5	0.171191	-0.000405±0.000349997	-0.10375±0.00116141
9	1	0.012315	-0.000237±0.000163739	-0.009953±0.000348387
9	1.5	0.001867	5e-06±7.01641e-05	-0.00156±0.000138312

4.5.2 Outliers

In this experiment, we analyze the impact of outliers in methods that do not consider the possibility of outliers. The purpose is to show that, although "max-prob" has a higher hit count than Greedy #2 when $q = 0$, if we increase q without changing the cost function of "max-prob" to incorporate outliers, then Greedy #2 rapidly outperforms "max-prob". On the other hand, if we use the correct cost function, i.e. incorporating the possibility of outliers with the correct value of q, we should obtain a higher hit count than Greedy #2.

We use again isotropic Gaussian distributions, this time with the generator set model and outliers. We fix $n = 2$ and $N = 30$ and vary $\epsilon/\sigma \in \{.01, .1, .25, .5, 1, 1.5\}$ and $q \in \{0, .1, , .2, ..., .9\}$, and pseudorandomly generate 10^3 different pairs P_1, P_2 and run "max-prob" without outliers, "max-prob" with outliers and Greedy #2 with them. The values of σ, ϵ and q are always known to the Bayesian methods, except for the method without outliers, which assumes $q = 0$, while Greedy #2 always uses Euclidean distance as cost. Figure 4.1 shows the hit count of the three algorithms in these conditions.

We note that, for low values of ϵ/σ, Greedy #2 and "max-prob" with outliers have similar results while "max-prob" without outliers rapidly deteriorates as q increases. For higher values of ϵ/σ, instead, "max-prob" with and without outliers yield approximately the same results, while Greedy #2 has lower hit counts.

4.5.3 Parameter robustness

This experiment evaluates parameter robustness, i.e., how much the hit rate deteriorates when one does not use the correct parameters (σ, ϵ and q in the isotropic Gaussian model with outliers).

We generated 10000 pairs of sets P_1, P_2 in the generator set model with isotropic Gaussian distributions with outliers, with $N = 30$, $\sigma = 1$, $\epsilon = .05^{1/n}$, $q = .4$ and $n \in \{1, 2, 5, 10\}$, and evaluated the hit count of "max-prob" with different parameter choices. In Figure 4.2(a) we choose the correct values for σ and ϵ and vary q. As one may see, hit count is very robust to the choice of q — as long as q is not too close to 0 or too close to 1, the hit count does not change much. If we choose the correct q and ϵ and vary σ (Figure 4.2(b)), similarly, as long as σ is not too close to 0, the hit count remains fairly stable. The most sensitive parameter seems to be ϵ: When we choose the correct values for σ and q and vary ϵ (Figure 4.2(c)), we notice that the hit count falls more dramatically as ϵ approaches zero than as σ or q do. We can also notice a slight concavity around the optimal value of ϵ (that is, around $\epsilon_{\text{formula}}/\epsilon_{\text{actual}} = 1$), which is not noticeable for the other parameters.

We conclude with this experiment that "max-prob" with outliers is quite robust

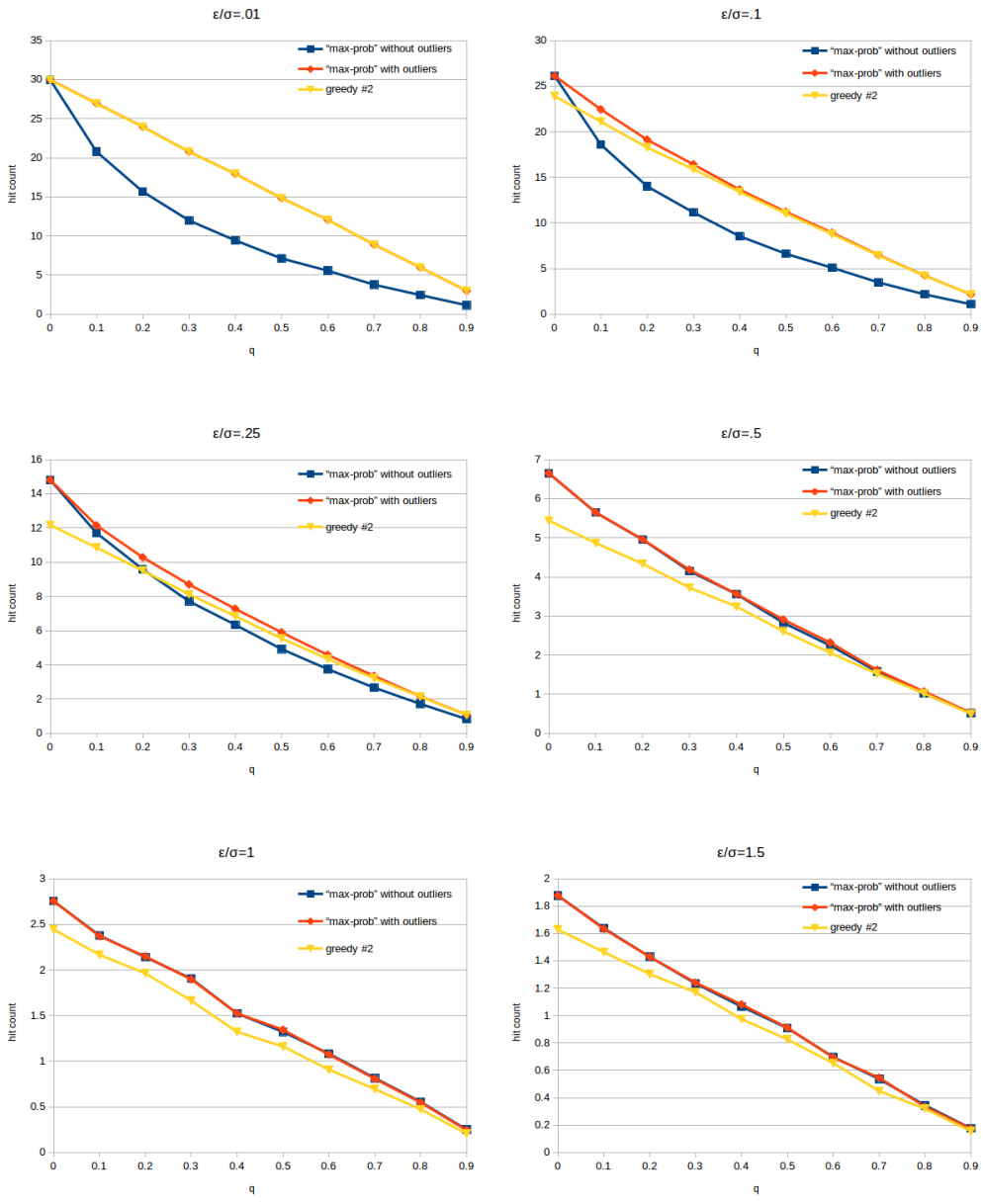
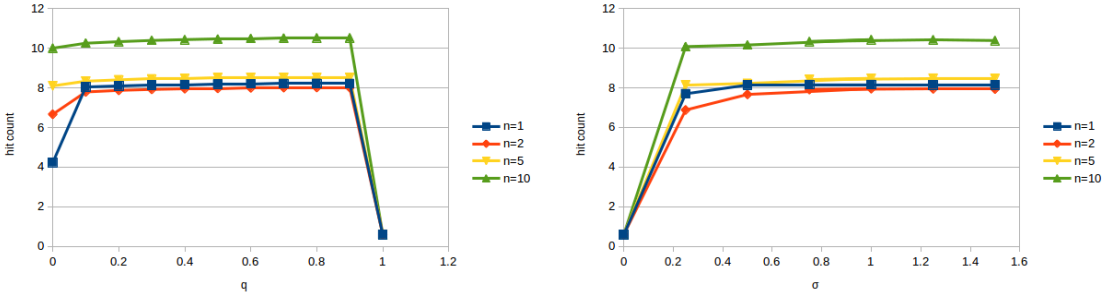
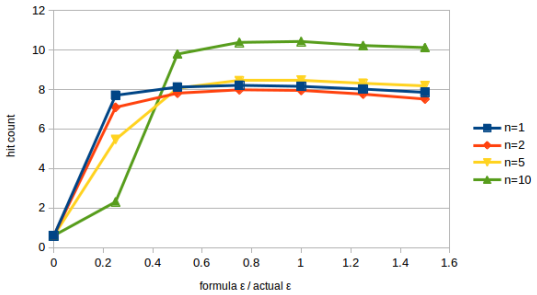


Figure 4.1: Hit count comparison of “max-prob” without outliers, “max-prob” with outliers and Greedy #2 in the direct model with outliers.



(a) varying q

(b) varying σ



(c) varying ϵ

Figure 4.2: Hit count as the parameters used in the cost function for "max-prob" differ from the actual parameters of the probabilistic model.

to parameter changes, where the most sensitive parameter is the noise parameter ϵ and the most robust one is the outlier rate q.

Part II

Theoretic Results

Chapter 5

Hit count of the "max-prob" problem

If we would like to analyze the behavior of our framework, it is reasonable to start with extreme cases.

If there is zero noise ($\epsilon = 0$ in the Gaussian model) for some fixed number of points N, it is certain that the "max-prob" algorithm will match correctly all points with 100% of probability. If there is infinite noise ($\sigma = 0$ in the Gaussian model, i.e. $\epsilon/\sigma = \infty$), there is no information that can be used for matching, so any method is as good as choosing a random permutation, which means that the probability of correctly matching all points is $1/N!$, and the expected hit count is equal to 1 (since the probability of correctly matching a given point $x_1 \in P_1$ is equal to $1/N$).

What about when we have an infinite number of points, for a fixed noise ratio? As $N \to \infty$, the sets become increasingly denser and it becomes gradually harder to match points correctly, so we know that the hit rate decreases with N. In this sense, increasing the number of points seems to have a similar effect to that of increasing the noise. Should we expect then that the expected hit count converges to 1 as $N \to \infty$, exactly as in the case when we have infinite noise?

If infinite noise gives us an expected hit count of 1 regardless of the value of N, it seems reasonable to think that infinitely many points but finite noise would give us better results than infinitely many points and infinite noise. Therefore, the expected hit count with infinitely many points and finite noise should be greater than or equal to 1.

In this chapter we compute this expected hit count with an infinite number of points. We will see that this value depends largely on the distribution of the generator set, being a finite number for Gaussian distributions, and infinite for heavy-tailed distributions such as power laws.

5.1 Infinitely many points

The expected hit count of "max-prob"[1], $E[\#\text{hits}_{\text{max-prob}}]$, can be written as:

$$E[\#\text{hits}_{\text{max-prob}}] = N \cdot P[\text{hit}_{\text{max-prob}}] = N \int_{\mathbb{R}^n} P[\text{hit}_{\text{max-prob}}|x_1]\text{pdf}[x_1]dx_1$$

where $P[\text{hit}_{\text{max-prob}}]$ is the probability of correctly matching a random point $x_1 \in P_1$, which has probability density of $\text{pdf}[x_1] = p_1(x_1)$.

Given only x_1, the match x_2^* found by the "max-prob" algorithm (whether it is correct or not) is located in a random position with distribution $\text{pdf}[x_2^*|x_1]$. We write then

$$E[\#\text{hits}_{\text{max-prob}}] = N \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} P[\text{hit}_{\text{max-prob}}|x_1, x_2^*]\text{pdf}[x_1]\text{pdf}[x_2^*|x_1]dx_1dx_2^*$$

To be able to compute this integral, we must do two substitutions. First of all, we must note that, as $N \to \infty$, $\text{pdf}[x_2^*|x_1]$ converges to a Dirac delta function (as we will show in Section 5.2), i.e., x_2^* becomes a function $x_2^*(x_1)$ when $N \to \infty$. Secondly, we will replace $P[\text{hit}_{\text{max-prob}}|x_1, x_2^*]$ with $P[\text{hit}|x_1, x_2]$, which refers to the probability that a given a pair (x_1, x_2) is a correct match[2]. These two are different probabilities, since the first case is also conditioned on the fact that x_2^* was provided by the "max-prob" algorithm (which increases the probability of the match being correct). However, since x_2^* converges to a Dirac delta, using $x_2 = x_2^*(x_1)$ will make $P[\text{hit}|x_1, x_2]$ converge to $P[\text{hit}_{\text{max-prob}}|x_1, x_2^*]$ as $N \to \infty$ (i.e. the fact that x_2 was generated by "max-prob" provides no extra information anymore, since x_2^* becomes deterministic). We will refrain from giving a formal proof of the correctness of this second substitution.

Applying the substitutions, we have:

$$\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] = \lim_{N\to\infty} N \int_{\mathbb{R}^n} P[\text{hit}|x_1, x_2]\text{pdf}[x_1]dx_1\Big|_{x_2=x_2^*(x_1)} \tag{5.1}$$

[1] In a more rigorous notation, $E[\#\text{hits}_{\text{max-prob}}] = E[\Pi_{\text{max-prob}} : \Pi]$, where $\Pi_{\text{max-prob}}$ is the solution of "max-prob", and Π is the correct permutation. Note that $\Pi_{\text{max-prob}}$ is a (deterministic) function $\Pi_{\text{max-prob}}(X_1, X_2)$, while X_1, X_2 and Π are random; but $\Pi_{\text{max-prob}}$ is random if X_1 or X_2 (or any parts of them) are not given. Similarly, the hit rate is $P[\text{hit}_{\text{max-prob}}] = P[(\pi_{\text{max-prob}})_i = \pi_i]$, for arbitrary i (e.g. $i = 1$; the choice of i has no loss of generality); and $P[\text{hit}_{\text{max-prob}}|x_1] = P[(\pi_{\text{max-prob}})_i = \pi_i|X_1^i = x_1]$. Note also that $E[\#\text{hits}_{\text{max-prob}}]$ and $P[\text{hit}_{\text{max-prob}}]$ depend on all model parameters, e.g. N, n, ϵ and σ in the case of the isotropic Gaussian model.

[2] In a more rigorous notation, $P[\text{hit}|x_1, x_2]$ means $P[\Pi_{ij} = 1|X_1^i = x_1, X_2^j = x_2]$. Note that x_2 and x_2^* are different concepts: while x_2 is a random point in P_2, x_2^* is the point "max-prob" matched x_1 to, i.e. $P[\text{hit}_{\text{max-prob}}|x_1, x_2^*] = P[\text{hit}|x_1, x_2, \text{"max-prob" matched } x_1 \text{ to } x_2]$. Note also that $P[\text{hit}|x_1, x_2]$ is different from the expression obtained in Equation 4.1: here, we are conditioning on individual points X_1^i and X_2^j, while Equation 4.1 conditions on the whole sets X_1 and X_2.

Now let us take a closer look into $P[\text{hit}|x_1, x_2]$. Using Bayes' rule we have

$$P[\text{hit}|x_1, x_2] = \frac{\text{pdf}[x_1, x_2|\text{hit}]P[\text{hit}]}{\text{pdf}[x_1, x_2]} = \frac{\text{pdf}[x_1, x_2|\text{hit}]P[\text{hit}]}{\text{pdf}[x_1, x_2|\text{hit}]P[\text{hit}] + \text{pdf}[x_1, x_2|\neg\text{hit}]P[\neg\text{hit}]}$$

Selecting two random points from P_1 and P_2 yields a correct match with probability $1/N$, so $P[\text{hit}] = 1/N$. Meanwhile, $\text{pdf}[x_1, x_2|\neg\text{hit}] = \text{pdf}[x_1|\neg\text{hit}]\text{pdf}[x_2|\neg\text{hit}] = \text{pdf}[x_1]\text{pdf}[x_2]$. Finally, $\text{pdf}[x_1, x_2|\text{hit}]$ is the joint probability $h(x_1, x_2)$ defined in Section 4.1.3. We obtain:

$$P[\text{hit}|x_1, x_2] = \frac{1/N}{1/N + \frac{N-1}{N} \cdot \frac{\text{pdf}[x_1]\text{pdf}[x_2]}{\text{pdf}[x_1,x_2|\text{hit}]}}$$

$$= \frac{1/N}{1/N + \frac{N-1}{N} \cdot \frac{p_1(x_1)p_2(x_2)}{h(x_1,x_2)}}$$

Replacing in Equation 5.1:

$$\begin{aligned}
\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] &= \lim_{N\to\infty} N \int_{\mathbb{R}^n} \frac{1/N}{1/N + \frac{N-1}{N} \cdot \frac{p_1(x_1)p_2(x_2^*(x_1))}{h(x_1,x_2^*(x_1))}} \cdot p_1(x_1)dx_1 \qquad (5.2)\\
&= \lim_{N\to\infty} \int_{\mathbb{R}^n} \frac{1}{1/N + \frac{N-1}{N} \cdot \frac{p_1(x_1)p_2(x_2^*(x_1))}{h(x_1,x_2^*(x_1))}} \cdot p_1(x_1)dx_1\\
&= \int_{\mathbb{R}^n} \frac{1}{\frac{p_1(x_1)p_2(x_2^*(x_1))}{h(x_1,x_2^*(x_1))}} \cdot p_1(x_1)dx_1\\
&= \int_{\mathbb{R}^n} \frac{h(x_1, x_2^*(x_1))}{p_2(x_2^*(x_1))} dx_1
\end{aligned}$$

Now to solve this integral we need to be able to calculate $x_2^*(x_1)$, which we will see in the next section.

5.2 Computing $x_2^*(x_1)$

If the number of dimensions is $n = 1$, supposing the points in P have a Gaussian distribution $p(x) = g_\sigma(x)$ and Gaussian noise $p_y(y) = g_\epsilon(y)$, then "max-prob" can be solved by sorting the sets, as seen in Section 4.1.5. Therefore, as $N \to \infty$, x_2^* converges to a Dirac delta function centered at $CDF_2^{-1}(CDF_1(x_1))$, where $CDF_1(a) = P[x_1 < a]$ and $CDF_2(a) = P[x_2 < a]$. If the direct model is being used, then $p_1(x_1) = g_\sigma(x_1)$ and $p_2(x_2) = g_{\sqrt{\sigma^2+\epsilon^2}}(x_2)$ implies that $x_2^*(x_1) = \frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} x_1$. If the generator set model is used, then $p_1(x) = p_2(x)$ and we have $x_2^*(x_1) = x_1$.

However, this approach only applies if $n = 1$ and distributions are Gaussian. Note also that this only applies to the "max-prob" algorithm: If for instance the greedy #2 algorithm (Section 2.1.2) is used instead of "max-prob", in the direct

model, the distribution of x_2^* will not converge to a Dirac delta function (as will be shown in Section 5.5.2).

In order to compute $x_2^*(x_1)$ in more general scenarios, we need to solve a variational calculus problem.

5.2.1 Nonlinear Variational Formulation

As N grows to infinity, the minimum bipartite matching problem acquires a completely different structure. Note that

- P_1 and P_2 become densely populated sets, eventually populating the whole domain (e.g. $\mathbb{R}^n$ in the case of Gaussian variables)[3]. As $N \to \infty$, the fraction of the points of P_1 contained in a region $\Omega \in \mathbb{R}^n$ will be exactly[4] $\int_\Omega p_1(x_1)dx_1$. The same applies to P_2.

- Because bipartite matching is one-to-one, if all the points of P_1 contained in a region $\Omega_1 \in \mathbb{R}^n$ are matched to all the points of P_2 contained in a region $\Omega_2 \in \mathbb{R}^n$, then these regions correspond to an equal fraction of the points in P_1 and P_2, i.e., $\int_{\Omega_1} p_1(x_1)dx_1 = \int_{\Omega_2} p_2(x_2)dx_2$. Therefore if $\text{pdf}[x_2^*|x_1]$ converges to a Dirac delta, then by reducing Ω_1 and Ω_2 to infinitesimal regions, we obtain $p_1(x_1)dx_1 = p_2(x_2)dx_2$, implying that $x_2^*(x_1)$ must satisfy[5] $|\det(\partial_{x_1} x_2^*(x_1))| = \frac{p_1(x_1)}{p_2(x_2^*(x_1))}$, where $\partial_{x_1} x_2^*(x_1)$ is the Jacobian of $x_2^*(x_1)$ (i.e., a matrix J such that $J_{ij} = \frac{\partial (x_2^*)_i}{\partial (x_1)_j}$).

- The cost of matching region Ω_1 to Ω_2, divided by N, would be equal to $\int_{\Omega_1} C(x_1, x_2^*(x_1))p_1(x_1)dx_1$, where $C(x_1, x_2)$ is the cost function (e.g. $-\log h(x_1, x_2)$).

From these properties we derive that, if $\text{pdf}[x_2^*|x_1]$ converges to a Dirac delta, then it must be the solution to the variational problem[6] below:

$$\min_{x_2^*:\mathbb{R}^n \mapsto \mathbb{R}^n} \int C(x_1, x_2^*(x_1))p_1(x_1)dx_1$$
$$\text{subject to: } |\det(\partial_{x_1} x_2^*(x_1))| = \frac{p_1(x_1)}{p_2(x_2^*(x_1))} \tag{5.3}$$

However, this formulation has two disadvantages: Firstly, it is it is highly nonlinear and therefore difficult to solve in this form; and secondly, it assumes $\text{pdf}[x_2^*|x_1]$ converges to a Dirac delta.

[3] That is, taking any small region $\Omega \subset \mathbb{R}^n$ with non-zero probability (i.e. $P[x_1 \in \Omega] > 0$), we have 100% probability of finding in Ω infinitely many points from P_1 or P_2 as $N \to \infty$ (i.e. $P\left[\lim_{N\to\infty} |P_1 \cap \Omega| = \infty\right] = 1$).

[4] derives from the law of large numbers.

[5] assuming $x_2^*(x_1)$ is smooth and invertible;

[6] In the literature this is known as Monge's formulation to the *optimal transport* problem [26].

5.2.2 Linear Variational Formulation

Suppose that $\text{pdf}[x_2^*|x_1]$ does not necessarily converge to a Dirac delta. That is, the points located in an infinitesimal region Ω_1 around a point $x_1 \in P_1$ are not necessarily matched to an infinitesimal region around a point $x_2^* \in P_2$, but rather to points anywhere around P_2 following some distribution function $\text{pdf}[x_2^*|x_1]$. Let then $\Pi(x_1, x_2^*) = \frac{\text{pdf}[x_2^*|x_1]}{p_2(x_2^*)}$. The cost we have to minimize is therefore[7]:

$$\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} C(x_1, x_2^*)\text{pdf}[x_2^*|x_1]p_1(x_1)dx_1dx_2^* = \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} C(x_1, x_2)\Pi(x_1, x_2)p_2(x_2)p_1(x_1)dx_1dx_2. \tag{5.4}$$

Because $\int_{\mathbb{R}^n} \text{pdf}[x_2^*|x_1]dx_2^* = 1$, it is clear that Π must satisfy:

$$\forall x_1 : \int_{\mathbb{R}^n} \Pi(x_1, x_2)p_2(x_2)dx_2 = 1 \tag{5.5}$$

Also, because matching is one-to-one, $\int_{\mathbb{R}^n} \text{pdf}[x_2^*|x_1]\text{pdf}[x_1]dx_1 = p_2(x_2^*)$, which leads us to:

$$\forall x_2 : \int_{\mathbb{R}^n} \Pi(x_1, x_2)p_1(x_1)dx_1 = 1 \tag{5.6}$$

And obviously, Π must be non-negative everywhere.

$$\Pi(x_1, x_2) \geq 0 \tag{5.7}$$

This variational problem[8] is strikingly similar to the primal problem of finite (fixed N) minimum bipartite matching (Section 2.2): The permutation matrix Π became a functional of two variables, and the matrix-vector products, as well as the matrix inner product, became functional inner products on the probability measures of sets P_1 and P_2.

Because this is a (linear) convex optimization problem, a candidate solution Π is optimal if and only if the Lagrange multipliers $u(x_1)$, $v(x_2)$ and $W(x_1, x_2)$ of the constraints of Equations 5.5, 5.6 and 5.7, respectively, satisfy the Karush-Kuhn-Tucker conditions of the problem[9]:

$$\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} W(x_1, x_2)\Pi(x_1, x_2)p_2(x_2)p_1(x_1)dx_1dx_2 = 0 \quad \text{(complementary slackness)}$$

$$W(x_1, x_2) \geq 0 \quad \text{(dual feasibility)}$$

$$W(x_1, x_2) + u(x_1) + v(x_2) = C(x_1, x_2) \quad \text{(stationarity)}$$

[7] We rename x_2^* to x_2 after Equation 5.4 for convenience.

[8] known as Kantorovich's formulation to the optimal transport problem [26];

[9] We simply rewrote the KKT conditions of the discrete version of the problem, i.e. $W : \Pi = 0$, $W \geq 0$ and $u\vec{1}^T + \vec{1}v^T + W = C$, in the variational framework, assuming that when $N \to \infty$ they remain valid and the Lagrange multipliers u, v and W converge to their respective functionals.

If we eliminate W substituting the third condition on the other two, we obtain:

$$\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} C(x_1,x_2)\Pi(x_1,x_2)p_2(x_2)p_1(x_1)dx_1dx_2 = ...$$
$$... \int_{\mathbb{R}^n}\int_{\mathbb{R}^n} (u(x_1)+v(x_2))\Pi(x_1,x_2)p_2(x_2)p_1(x_1)dx_1dx_2; \tag{5.8}$$
$$u(x_1)+v(x_2) \leq C(x_1,x_2). \tag{5.9}$$

Equations 5.5 and 5.6 imply that Equation 5.8 can be simplified to:

$$\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} C(x_1,x_2)\Pi(x_1,x_2)p_2(x_2)p_1(x_1)dx_1dx_2 = ...$$
$$... \int_{\mathbb{R}^n} u(x_1)p_1(x_1)dx_1 + \int_{\mathbb{R}^n} v(x_2)p_2(x_2)dx_2 \tag{5.10}$$

Now, if pdf$[x_2^*|x_1]$ is a Dirac delta, then we must have $\Pi(x_1,x_2) = \frac{\delta(x_2-x_2^*(x_1))}{p_2(x_2)}$, where δ is the Dirac delta function in $\mathbb{R}^n$. In this case Π satisfies the constraints of Equations 5.5 and 5.6 because:

$$\int_{\mathbb{R}^n} \Pi(x_1,x_2)p_2(x_2)dx_2 = \int_{\mathbb{R}^n} \delta(x_2-x_2^*(x_1))dx_2 = 1$$

and, supposing $x_2^*(x_1)$ satisfies Equation 5.3:

$$\int_{\mathbb{R}^n} \Pi(x_1,x_2)p_1(x_1)dx_1 = \int_{\mathbb{R}^n} \delta(x_2-x_2^*(x_1))\frac{p_1(x_1)}{p_2(x_2^*(x_1))}dx_1 = \int_{\mathbb{R}^n} \delta(x_2-x_2^*)dx_2^* = 1.$$

And in this case, Equation 5.10 becomes

$$\int_{\mathbb{R}^n} C(x_1,x_2^*(x_1))p_1(x_1)dx_1 = \int_{\mathbb{R}^n} (u(x_1)+v(x_2^*(x_1)))p_1(x_1)dx_1$$

which, because $u(x_1)+v(x_2^*(x_1)) \leq C(x_1,x_2^*(x_1))$ (Equation 5.9), is only satisfied if

$$\forall x_1 : u(x_1)+v(x_2^*(x_1)) = C(x_1,x_2^*(x_1)).$$

Therefore, to test whether a particular function $x_2^*(x_1)$ is the solution[10] of "max-prob" as $N\to\infty$, i.e. whether pdf$[x_2^*|x_1] = \delta(x_2^*-x_2^*(x_1))$ for this particular choice of $x_2^*(x_1)$; we have to find u,v that satisfy:

$$\forall x_1,x_2 : u(x_1)+v(x_2) \leq C(x_1,x_2) \tag{5.11}$$
$$\forall x_1 : u(x_1)+v(x_2^*(x_1)) = C(x_1,x_2^*(x_1)) \tag{5.12}$$

[10]It is important to note however that this methodology does not prove the uniqueness of the solution. That is, the global minimum may well be a continuum of solutions. In other words, even if we prove that $\Pi(x_1,x_2) = \frac{\delta(x_2-x_2^*(x_1))}{p_2(x_2)}$ for a given $x_2^*(x_1)$ minimizes the total cost, there may be other functionals $\Pi(\cdot,\cdot)$ (and not necessarily in a Dirac delta form) that have the same cost.

Solution

From Equation 5.12 we derive that $\forall a : v(a) = C(x_2^{*-1}(a), a) - u(x_2^{*-1}(a))$, which, substituting in Equation 5.11, results in:

$$\forall x_1, x_2 : u(x_1) + C(x_2^{*-1}(x_2), x_2) - u(x_2^{*-1}(x_2)) \leq C(x_1, x_2)$$

which, rewriting $x_1 = a$ and $x_2 = x_2^*(b)$, becomes:

$$\begin{aligned} &\forall a, b : u(a) + C(b, x_2^*(b)) - u(b) \leq C(a, x_2^*(b)) \\ \Leftrightarrow\ &\forall a, b : u(a) - C(a, x_2^*(b)) \leq u(b) - C(b, x_2^*(b)) \\ \Leftrightarrow\ &\forall b : b \in \arg\max_a \left\{u(a) - C(a, x_2^*(b))\right\}. \end{aligned} \tag{5.13}$$

Ultimately, we need to find u that satisfies Equation 5.13 for this choice of $x_2^*(x_1)$. The maximum argument occurs where the gradient of the expression is zero, and therefore Equation 5.13 implies:

$$\begin{aligned} &\partial_{x_1} u(b) - \partial_{x_1} C(b, x_2^*(b)) = 0 \\ \Leftrightarrow\ &\partial_{x_1} u(b) = \partial_{x_1} C(b, x_2^*(b)) \end{aligned} \tag{5.14}$$

So we can pursue u by solving this differential equation. This is the classical problem of finding the potential of a vector field: The equation

$$\nabla u(x) = F(x) \tag{5.15}$$

for $u : \mathbb{R}^n \to \mathbb{R}$ and $F : \mathbb{R}^n \to \mathbb{R}^n$ has a solution if and only if the Jacobian $\partial_x F(x)$ is a symmetric matrix for all x.

This condition must be satisfied because, if we further derive this equation, obtaining

$$\nabla^2 u(x) = \partial_x F(x),$$

where ∇^2 denotes the Hessian operator, the fact that a Hessian matrix is always symmetric implies that $\partial_x F(x)$ must be symmetric.

Meanwhile, if $\partial_x F$ is a symmetric matrix, then the work (line integral) of any path linking a constant point (for instance, the origin, 0) to x is a solution[11] to u. So for instance,

$$u(x) = \int_{t=0}^{1} \langle F(tx), d(tx) \rangle = \int_0^1 \langle x, F(tx) \rangle dt$$

[11] There are infinitely many solutions to u: Because $\nabla \{u(x) + const.\} = \nabla u(x)$, both $u(x)$ and $\tilde{u}(x) = u(x) + const.$ are solutions to $\nabla u(x) = F(x)$.

is a solution to Equation 5.15, since

$$\nabla u(x) = \int_0^1 \nabla_x \langle x, F(tx) \rangle dt$$

$$= \int_0^1 F(tx) + t\partial_x F(tx)^T x dt,$$

which, applying the product rule $\int_0^1 uv'dt = uv|_0^1 - \int_0^1 u'vdt$ with $u = F(tx)$ and $v = t$ to the first term of the integrand, becomes

$$\begin{aligned}
\nabla u &= tF(tx)|_{t=0}^1 + \int_0^1 -t\partial_x F(tx)x + t\partial_x F(tx)^T x dt \\
&= F(x) + \int_0^1 t(\partial_x F(tx)^T - \partial_x F(tx))x dt \\
&= F(x),
\end{aligned}$$

since $\partial_x F(tx)^T - \partial_x F(tx) = 0$.

In our case[12], $F(x_1) = \partial_{x_1} C(x_1, x_2^*(x_1))^T \Rightarrow \partial_x F(x_1) = \partial_{x_1}\partial_{x_1} C(x_1, x_2^*(x_1)) + \partial_{x_2}\partial_{x_1} C(x_1, x_2^*(x_1))\partial_{x_1} x_2^*(x_1)$, where $(\partial_{x_2}\partial_{x_1} C(x_1, x_2))_{i,j} = \partial_{(x_2)_j}\partial_{(x_1)_i} C(x_1, x_2)$. As $\partial_{x_1}\partial_{x_1} C(x_1, x_2^*(x_1))$ is already symmetric, we only need to verify if $\partial_{x_2}\partial_{x_1} C(x_1, x_2^*(x_1))\partial_{x_1} x_2^*(x_1)$ is symmetric.

Furthermore, since this is a maximization problem (Equation 5.13), the Hessian $\nabla_a^2 \{u(a) - C(a, x_2^*(b))\}|_{a=b} = \nabla^2 u(b) - \partial_{x_1}\partial_{x_1} C(b, x_2^*(b)) = \partial_{x_2}\partial_{x_1} C(b, x_2^*(b))\partial_{x_1} x_2^*(b)$ must be negative semidefinite for all b.

To summarize, the method to test whether a mapping function $x_2^*(x_1)$ is the solution provided by "max-prob" as $N \to \infty$, we need to:

1. Check if $\forall x_1 : |\det(\partial_{x_1} x_2^*(x_1))| = p_1(x_1)/p_2(x_2^*(x_1))$;
2. Check if the matrix $\partial_{x_2}\partial_{x_1} C(b, x_2^*(b))\partial_{x_1} x_2^*(b)$ is symmetric for all b;
3. Calculate $u(x_1) = \int_0^1 \langle \partial_{x_1} C(tx_1, tx_2^*(x_1))^T, x_1 \rangle dt$;
4. Check if $\forall b : b \in \arg\max_a \{u(a) - C(a, x_2^*(b))\}$. Note that this will only be possible if the matrix from step 2 is also negative semidefinite.

Notice that the invariance properties of the cost function for the minimum bipartite matching problem are still valid at this point: If we change $C(x, y)$ to $\tilde{C}(x, y) = \alpha C(x, y) + f(x) + g(y)$, then $\tilde{u}(x) = \alpha u(x) + f(x)$ will satisfy Equation 5.13, therefore the solution to the matching problem will be the same mapping function $x_2^*(x_1)$.

[12]The transpose in "$\partial_{x_1} C^T$" denotes that this is a column vector, as we follow the convention that $(\partial_x y)_{ij} = \partial_{x_j} y_i$; i.e. if $y(x)$ is $y : \mathbb{R}^n \to \mathbb{R}$, then $\partial_x y$ is a row vector.

5.2.3 Simple examples

Before solving the cases we are interested in, let us see how this method can be used to solve some simple examples.

Squared Euclidean distance cost, same distribution

Consider that P_1 and P_2 have isotropic Gaussian distributions with parameter σ and the cost function is $C(x_1, x_2) = ||x_1 - x_2||^2$. The solution is obviously $x_2^*(x_1) = x_1$, since the probability densities are equal in P_1 and P_2 and this mapping would have zero cost.

1. We can see that it satisfies the constraint of Equation 5.3 as $\det \partial_{x_1} x_2^*(x_1) = \det I = 1 = p_1(x_1)/p_2(x_1)$.

2. $\partial_{x_2}\partial_{x_1} C(x_1, x_2) = -2I$ and $\partial_{x_1} x_2^*(x_1) = I$, so the symmetry constraint is satisfied. We also verify that the Hessian matrix is negative definite.

3. $\partial_{x_1} C(x_1, x_2)^T = 2(x_1 - x_2)$ implies that $u(x_1) = \int_0^1 \langle \partial_{x_1} C(tx_1, tx_2^*(x_1))^T, x_1 \rangle dt$ $= \int_0^1 \langle 2(tx_1 - tx_1), x_1 \rangle dt = 0$ for all x_1.

4. $u(a) - C(a, x_2^*(b)) = 0 - ||a - b||^2$ has its maximum when $a = b$.

Therefore, $x_2^*(x_1) = x_1$ is the correct solution to this case. On the other hand, if we test instead the mapping $x_2^*(x_1) = Qx_1$ for some orthogonal matrix Q, we still satisfy the constraint of Equation 5.3, because $|\det Q| = 1$ and $p_1(x_1) = p_2(Qx_1)$, but there exists no u that satisfies Equation 5.13, since

$$\partial_{x_2}\partial_{x_1} C(b, x_2^*(b))\partial_{x_1} x_2^*(b) = -2Q$$

is not a symmetric matrix. Similarly, if we tested $x_2^*(x_1) = -x_1$, we would satisfy the probability measure preservation constraint and the Hessian matrix would be symmetric, but positive definite, not negative definite, so step 4 would not be satisfied, although we would be able to compute $u(x_1) = 2||x_1||^2$.

Squared Euclidean distance cost, translated distribution

Let us see another example. Consider again the Gaussian distributions with variance parameter σ, but centered at different positions: $E[x_1] = 0$ and $E[x_2] = d$, for some $d \in \mathbb{R}^n$. Also, suppose the cost function is the same: $C(x_1, x_2) = ||x_1 - x_2||^2$. In this case, it seems reasonable to try $x_2^*(x_1) = x_1 + d$.

1. It satisfies Equation 5.3 as $\det \partial_{x_1} x_2^*(x_1) = \det I = 1 = p_1(x_1)/p_2(x_2^*(x_1))$;

2. $\partial_{x_2}\partial_{x_1}C(x_1,x_2) = -2I$ and $\partial_{x_1}x_2^*(x_1) = I$, so the symmetry constraint is satisfied. We also verify that the Hessian matrix is negative definite.

3. $u(x_1) = \int_0^1 \langle \partial_{x_1} C(tx_1, tx_2^*(x_1))^T, x_1\rangle dt = \int_0^1 \langle 2(tx_1 - tx_1 - d), x_1\rangle dt = -2\langle d, x_1\rangle$

4. In this case, $u(a) - C(a, x_2^*(b)) = -2\langle a, d\rangle - ||a - b - d||^2$. Deriving, we have $\nabla_a \{u(a) - C(a, x_2^*(b))\} = -2d - 2(a - b - d) = -2(a - b)$, so the expression has only one critical point, which is where $a = b$. It is the global maximum as the Hessian matrix was already verified to be negative definite.

5.2.4 Direct model case

Gaussian distributions, no outliers

Let us see now the solution to a more generic Gaussian case. In the direct model with isotropic Gaussian noise, the cost function is the squared Euclidean distance, as seen in Section 4.1.1. Suppose the points in P_1 have a Gaussian distribution with variance matrix Σ_1^2 and P_2 therefore with variance $\Sigma_2^2 = \Sigma_1^2 + \epsilon^2 I$. To simplify notation, let the "2" exponent refer to the products $\Sigma_1\Sigma_1^T$ and $\Sigma_2\Sigma_2^T$, and "-2" refer to $\Sigma_1^{-T}\Sigma_1^{-1}$ and $\Sigma_2^{-T}\Sigma_2^{-1}$, so that Σ_1 and Σ_2 do not need to be symmetric matrices.

First of all, we need to find a candidate transformation $x_2^*(x_1)$ that preserves the probability measure, i.e. $|\det(\partial_{x_1}x_2^*(x_1))| = \frac{p_1(x_1)}{p_2(x_2^*(x_1))}$. We can show that a linear transformation in the form $x_2^*(x_1) = Tx_1$, where $T = \Sigma_2 Q \Sigma_1^{-1}$, for any orthogonal matrix Q, preserves the probability measure:

$$\begin{aligned}
p_1(x_1)/p_2(x_2^*(x_1)) &= \frac{e^{-\frac{1}{2}x_1^T\Sigma_1^{-2}x_1}/((2\pi)^{n/2}\det\Sigma_1)}{e^{-\frac{1}{2}x_2^*(x_1)^T\Sigma_2^{-2}x_2^*(x_1)}/((2\pi)^{n/2}\det\Sigma_2)} \\
&= \frac{\det\Sigma_2}{\det\Sigma_1}\exp\left(-\frac{1}{2}x_1^T\Sigma_1^{-2}x_1 + \frac{1}{2}x_2^*(x_1)^T\Sigma_2^{-2}x_2^*(x_1)\right) \\
&= \frac{\det\Sigma_2}{\det\Sigma_1}\exp\left(-\frac{1}{2}x_1^T\Sigma_1^{-2}x_1 + \frac{1}{2}x_1^T\Sigma_1^{-T}Q^T\Sigma_2^T\Sigma_2^{-2}\Sigma_2 Q\Sigma_1^{-1}x_1\right) \\
&= \frac{\det\Sigma_2}{\det\Sigma_1}\exp\left(-\frac{1}{2}x_1^T\Sigma_1^{-2}x_1 + \frac{1}{2}x_1^T\Sigma_1^{-2}x_1\right) \\
&= \frac{\det\Sigma_2}{\det\Sigma_1} \\
&= |\det(\partial_{x_1}\{\Sigma_2 Q\Sigma_1^{-1}x_1\})|
\end{aligned}$$

Step 2. As $\partial_{x_2}\partial_{x_1}C(x_1,x_2) = -2I$, the Hessian is:

$$\partial_{x_2}\partial_{x_1}C(b, x_2^*(b))\partial_{x_1}x_2^*(b) = -2T$$

which means that T must be symmetric positive definite.

Step 3. The calculation of u is straightforward:

$$u(x_1) = \int_0^1 \langle x_1, 2(tx_1 - x_2^*(tx_1))\rangle dt = \int_0^1 2t\langle x_1, x_1 - Tx_1\rangle dt = x_1^T(I-T)x_1$$

Step 4. Now let us analyze $u(a) - C(a, x_2^*(b))$:

$$u(a) - C(a, x_2^*(b)) = a^T(I-T)a - ||a - Tb||^2$$

Its derivative is:

$$2(I-T)a - 2(a - Tb) = -2Ta + 2Tb$$

implying that the only critical point occurs where $a = b$, which is the global maximum if T is positive definite.

Now we only need to show that $T = \Sigma_2 Q \Sigma_1^{-1}$ is symmetric positive definite. In fact, this is only happens for one particular choice of Q.

We can write:

$$T = \Sigma_2 Q \Sigma_1^{-1} = \Sigma_1^{-T}(\Sigma_1^T \Sigma_2 Q)\Sigma_1^{-1}$$

Now let $UDV^T = \Sigma_2^T \Sigma_1$ be the singular value decomposition of $\Sigma_2^T \Sigma_1$. We obtain:

$$T = \Sigma_1^{-T}(VDU^T Q)\Sigma_1^{-1}$$

By choosing $Q = UV^T$, we get $T = \Sigma_1^{-T}(VDV^T)\Sigma_1^{-1}$ and T is therefore positive definite. It is proved now that $x_2^*(x_1) = Tx_1$ is solution of "max-prob" as $N \to \infty$ in the direct model with Gaussian distributions, for this particular choice of T.

Gaussian distributions with outliers

Suppose now we have again the direct model with Gaussian distributions of variance Σ_1^2 and $\Sigma_2^2 = \Sigma_1^2 + \epsilon^2 I$, this time with outliers. Let us use normalized cost function #2 (Section 4.1.3):

$$C(x_1, x_2) = -\log \tilde{\zeta}(x_1, x_2) = -\log(q + (1-q)\zeta(x_1, x_2)) =$$

$$-\log\left(q + (1-q)\frac{h(x_1, x_2)}{p_1(x_1)p_2(x_2)}\right) = -\log\left(q + (1-q)\frac{\text{pdf}[x_2|x_1, \text{inlier}]}{p_2(x_1)}\right) =$$

$$-\log\left(q + (1-q)\frac{p_y(x_2 - x_1)}{p_2(x_2)}\right) = -\log\left(q + \frac{(1-q)\det\Sigma_2}{\epsilon^n} \cdot e^{-\frac{||x_1 - x_2||^2}{2\epsilon^2} + \frac{1}{2}||x_2||^2_{\Sigma_2^{-2}}}\right)$$

Because the prior distributions $p_1(x_1)$ and $p_2(x_2)$ are the same as the case without outliers, it is reasonable to try the same mapping function $x_2^*(x_1) = Tx_1$. Step 1 is automatically verified.

Step 2. First of all:

$$\partial_{x_1} C(x_1,x_2)^T = -\partial_{x_1}\left(\log\left(q+(1-q)\zeta(x_1,x_2)\right)\right)^T = -\frac{(1-q)\partial_{x_1}\zeta(x_1,x_2)^T}{q+(1-q)\zeta(x_1,x_2)}$$

$$= \frac{x_1-x_2}{\epsilon^2}\cdot\frac{(1-q)\zeta(x_1,x_2)}{q+(1-q)\zeta(x_1,x_2)} = \frac{x_1-x_2}{\epsilon^2}\cdot\frac{1}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1}.$$

Secondly,

$$\partial_{x_2}\partial_{x_1} C(x_1,x_2) = \frac{-I/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^2\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q\partial_{x_2}\zeta(x_1,x_2)^T}{(1-q)\zeta(x_1,x_2)^2}$$

$$= \frac{-I/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^2\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q/(1-q)}{\zeta(x_1,x_2)}\left(\frac{(x_1-x_2)^T}{\epsilon^2}+x_2\Sigma_2^{-2}\right)$$

$$= \frac{-I/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^2\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q/(1-q)}{\zeta(x_1,x_2)}\left(\frac{x_1^T}{\epsilon^2}-\frac{x_2^T(\Sigma_2^2-\epsilon^2 I)\Sigma_2^{-2}}{\epsilon^2}\right)$$

$$= \frac{-I/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^4\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q/(1-q)}{\zeta(x_1,x_2)}\left(x_1^T-x_2^T\Sigma_1^2\Sigma_2^{-2}\right)$$

Because $\Sigma_2^2=\Sigma_1^2+\epsilon^2 I$, we know that Σ_2^2 and Σ_1^2 have the same eigenvectors, and $T=\Sigma_2 Q\Sigma_1^{-1}$ also has the same eigenvectors: If we choose[13] Σ_1 and Σ_2 as the symmetric square roots of Σ_1^2 and Σ_2^2, then they also have the same eigenvectors and consequently the singular value decomposition $UDV^T=\Sigma_2^T\Sigma_1$ is also a symmetric matrix, implying that $U=V$, so $Q=I$ and T has the same eigenvectors as Σ_1^2 and Σ_2^2. Therefore, we know that $\Sigma_1^2\Sigma_2^{-2}=T^{-2}$, obtaining:

$$\partial_{x_2}\partial_{x_1} C(x_1,x_2) = \frac{-I/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^4\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q/(1-q)}{\zeta(x_1,x_2)}\left(x_1^T-x_2^T T^{-2}\right)$$

So,

$$\partial_{x_2}\partial_{x_1} C(x_1,x_2)\partial_{x_1} x_2^*(x_1) = \frac{-T/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,x_2)}+1} + \frac{(x_1-x_2)}{\epsilon^4\left(\frac{q/(1-q)}{\zeta(x_1,x_2)}+1\right)^2}\cdot\frac{q\left(x_1^T-x_2^T T^{-2}\right)T}{(1-q)\zeta(x_1,x_2)} \Rightarrow$$

[13]The choice of the matrix square root method should not impact the resulting matrix $T=\Sigma_2 Q\Sigma_1^{-1}$ if Q is obtained using the SVD method as described in the previous section: Suppose we use some other matrix Σ_1' instead of Σ_1 for the square root of Σ_1^2. Then $\Sigma_1'=\Sigma_1 Q_1$, for some orthogonal matrix Q_1 (i.e., if $AA^T=BB^T$, then $A^{-1}BB^TA^{-T}=I$, implying $A^{-1}B$ is orthogonal). Consider the same for Σ_2'. Then the new T is $T'=\Sigma_2' Q'{\Sigma_1'}^{-1}$, where $Q'=U'V'^T$ and $U'D'V'^T={\Sigma_2'}^T\Sigma_1'=Q_2^T\Sigma_2^T\Sigma_1 Q_1$. Therefore $D'=D$, $U'=Q_2^T U$, $V'=Q_1^T V$, $Q'=Q_2^T Q Q_1$ and $T'=\Sigma_2 Q_2 Q_2^T Q Q_1 Q_1^T\Sigma_1^{-1}=\Sigma_2 Q\Sigma_1^{-1}=T$ remains unchanged.

$$\partial_{x_2}\partial_{x_1}C(x_1,x_2^*(x_1))\partial_{x_1}x_2^*(x_1)=\frac{-T/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,Tx_1)}+1}+\frac{(x_1-Tx_1)}{\epsilon^4\left(\frac{q/(1-q)}{\zeta(x_1,Tx_1)}+1\right)^2}\cdot\frac{q\left(x_1^T-x_1^TTT^{-2}\right)T}{(1-q)\zeta(x_1,Tx_1)}$$

$$=\frac{-T/\epsilon^2}{\frac{q/(1-q)}{\zeta(x_1,Tx_1)}+1}-\frac{q(I-T)x_1x_1^T(I-T)}{\epsilon^4(1-q)\left(\frac{q/(1-q)}{\zeta(x_1,Tx_1)}+1\right)^2\zeta(x_1,Tx_1)}$$

which is always symmetric negative definite.

Step 3. $u(x_1)$ is equal the following integral:

$$u(x_1)=\frac{x_1(I-T)x_1}{\epsilon^2}\int_0^1\frac{tdt}{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(-\frac{1}{2}x_1^T\left(T\Sigma_2^{-2}T-\frac{(I-T)^2}{\epsilon^2}\right)x_1t^2\right)}$$

As shown before, we can use that $\Sigma_2^{-2}-I/\epsilon^2=-\Sigma_1^2\Sigma_2^{-2}/\epsilon^2=-T^{-2}/\epsilon^2$, obtaining:

$$T\Sigma_2^{-2}T-\frac{(I-T)^2}{\epsilon^2}=T(\Sigma_2^{-2}-I/\epsilon^2)T-\frac{I-2T}{\epsilon^2}=2\frac{T-I}{\epsilon^2}$$

so that

$$u(x_1)=\frac{x_1(I-T)x_1}{\epsilon^2}\int_0^1\frac{tdt}{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1t^2\right)}.$$

Now using that

$$\int\frac{1}{ae^{\lambda x}+b}dx=\frac{x}{b}-\frac{1}{\lambda b}\log\left(ae^{\lambda x}+b\right)+const.,$$

we obtain

$$u(x_1)=\frac{x_1^T(I-T)x_1}{\epsilon^2}\left[\frac{t^2}{2}-\frac{\log\left(1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1t^2\right)\right)}{2x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1}\right]_{t=0}^1$$

$$=\frac{x_1^T(I-T)x_1}{2\epsilon^2}\left(1-\frac{\log\left(\frac{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1\right)}{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}}\right)}{x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1}\right)$$

$$=\frac{x_1^T(I-T)x_1}{2\epsilon^2}-\frac{1}{2}\log\left(\frac{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1\right)}{1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}}\right)$$

or, because $u(x_1)$ is invariant to adding a constant, simply:

$$u(x_1)=\frac{x_1^T(I-T)x_1}{2\epsilon^2}-\frac{1}{2}\log\left(1+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\exp\left(x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1\right)\right)$$

$$=-\frac{1}{2}\log\left(e^{-x_1^T\left(\frac{I-T}{\epsilon^2}\right)x_1}+\frac{q\epsilon^n}{(1-q)\det\Sigma_2}\right)$$

$$= \frac{1}{2}C(x_1, x_2^*(x_1)) + const.$$

So a solution to u is $u(x_1) = \frac{1}{2}C(x_1, x_2^*(x_1))$.

Step 4. Let

$$U(a,b) \triangleq \exp(-C(a,Tb)) = \frac{(1-q)\det\Sigma_2}{\epsilon^n} \cdot e^{-\frac{||a-Tb||^2}{\epsilon^2} + \frac{1}{2}||Tb||^2_{\Sigma_2^{-2}}} + q$$

$$\triangleq \alpha e^{f(a,b)} + \beta,$$

so that $u(x_1) = -\log(U(x_1, x_1))/2$. We have to show that

$$\forall a, b : u(b) - C(b, x_2^*(b)) \geq u(a) - C(a, x_2^*(b))$$

$$\Leftrightarrow \forall a, b : -\frac{1}{2}\log U(b,b) + \log U(b,b) \geq -\frac{1}{2}\log U(a,a) + \log U(a,b)$$

$$\Leftrightarrow \forall a, b : \frac{1}{2}\log U(a,a) + \frac{1}{2}\log U(b,b) \geq \log U(a,b)$$

$$\Leftrightarrow U(a,a)U(b,b) \geq U(a,b)^2$$

$$\Leftrightarrow \alpha^2 e^{f(a,a)+f(b,b)} + \alpha\beta\left(e^{f(a,a)} + e^{f(b,b)}\right) + \beta^2 \geq \alpha^2 e^{2f(a,b)} + 2\alpha\beta e^{f(a,b)} + \beta^2$$

If it sufficient to prove that:

$$\begin{cases} f(a,a) + f(b,b) \geq 2f(a,b) \\ e^{f(a,a)} + e^{f(b,b)} \geq 2e^{f(a,b)} \end{cases}$$

Note however that because e^x is a convex function, i.e., $\forall x, y : e^{\frac{x+y}{2}} \leq \frac{e^x + e^y}{2}$, the first condition automatically implies the second, so we only need to prove the first one.

$$f(a,a) + f(b,b) \geq 2f(a,b) \Leftrightarrow$$

$$-\frac{||a-Ta||^2}{2\epsilon^2} + \frac{1}{2}||Ta||^2_{\Sigma_2^{-2}} - \frac{||b-Tb||^2}{2\epsilon^2} + \frac{1}{2}||Tb||^2_{\Sigma_2^{-2}} \geq -\frac{||a-Tb||^2}{\epsilon^2} + ||Tb||^2_{\Sigma_2^{-2}}$$

Using, as shown before, that $T^2/\epsilon^2 - T\Sigma_2^{-2}T = I/\epsilon^2$ we have

$$\frac{a^T(T-I)a}{\epsilon^2} + \frac{b^T(T-I)b}{\epsilon^2} \geq -\frac{||a||^2}{\epsilon^2} + 2\frac{a^TTb}{\epsilon^2} - \frac{||b||^2}{\epsilon^2} \Leftrightarrow$$

$$\frac{(a-b)^TT(a-b)}{\epsilon^2} \geq 0$$

which is true because T is positive definite.

Isotropic distributions, no outliers

In the direct model with Gaussian noise and no outliers, cost function is the

Euclidean distance squared, regardless of the distribution of the points in P_1. Let us consider the case that the distribution of P_1 is isotropic, i.e. $p_1(x_1)$ is actually $p_1(||x_1||)$ (by abuse of notation).

In this case, a reasonable mapping is $x_2^*(x_1) = R_2^{-1}(R_1(||x_1||))\frac{x_1}{||x_1||}$, where $R_1(t) \triangleq P[||x_1|| < t]$ and $R_2(t) \triangleq P[||x_2|| < t]$.

Step 1. First of all, note that

$$R_k'(||x_k||) = A_n||x_k||^{n-1}p_k(x_k),\ k \in \{1, 2\}$$

where A_n is the $(n-1)$-dimensional hyper-area of the boundary of an n-dimensional hyper-sphere of radius 1 (see Appendix C). Therefore,

$$\frac{p_1(x_1)}{p_2(x_2)} = \frac{||x_2||^{n-1}R_1'(||x_1||)}{||x_1||^{n-1}R_2'(||x_2||)}.$$

The Jacobian of $x_2^*(x_1)$ is then

$$\partial_{x_1}x_2^*(x_1) = R_2^{-1}(R_1(||x_1||))\left(\frac{I - \frac{x_1x_1^T}{||x_1||^2}}{||x_1||}\right) + \frac{R_1'(||x_1||)}{R_2'(R_2^{-1}(R_1(||x_1||)))} \cdot \frac{x_1x_1^T}{||x_1||^2}$$

which is a symmetric positive semidefinite matrix, with

$$\det \partial_{x_1}x_2^*(x_1) = \left(\frac{R_2^{-1}(R_1(||x_1||))}{||x_1||}\right)^{n-1} \frac{R_1'(||x_1||)}{R_2'(R_2^{-1}(R_1(||x_1||)))} = \frac{p_1(x_1)}{p_2(x_2^*(x_1))}.$$

Step 2. $\partial_{x_2}\partial_{x_1}C = -2I$, and $\partial_{x_1}x_2^*(x_1)$ is symmetric positive semidefinite, so this step is automatically verified.

Step 3. We do not need to compute u to show step 4 this time, so we skip step 3.

Step 4. It is sufficient to show that $u(a) - C(a, x_2^*(b))$ is concave with respect to a, i.e. the Hessian matrix $\nabla_a^2\{u(a) - C(a, x_2^*(b))\}$ is negative semidefinite for all a, b (not only when $a = b$):

$$\nabla_a^2\{u(a) - C(a, x_2^*(b))\} = \nabla^2 u(a) - \partial_{x_1}\partial_{x_1}C(a, x_2^*(b))$$

$$= \partial_{x_1}\partial_{x_1}C(a, x_2^*(a)) + \partial_{x_2}\partial_{x_1}C(a, x_2^*(a))\partial_{x_1}x_2^*(a) - \partial_{x_1}\partial_{x_1}C(a, x_2^*(b))$$

$$= 2I - 2\partial_{x_1}x_2^*(a) - 2I = -2\partial_{x_1}x_2^*(a),$$

which is negative semidefinite for all a.

5.2.5 Generator set model case

In the generator set model, the fact that $p_2(x) = p_1(x)$ makes the functional $x_2^*(x_1) = x_1$ the natural candidate solution. We will show that it is the solution of "max-prob" for any generator set distribution and any noise distribution.

Step 1. The candidate solution is feasible since $\det \partial_{x_1} x_2^*(x_1) = \det I = 1 = p_1(x_1)/p_2(x_1)$.

Step 2. Because $C(x_1, x_2)$ is a symmetric function, we know that $\partial_{x_1}\partial_{x_2} C(b, b)$ is a symmetric matrix, which satisfies the symmetry constraint[14].

Step 3. Noting that

$$\frac{d}{dt}C(tb, tb) = \partial_{x_1} C(tb, tb)b + \partial_{x_2} C(tb, tb)b = 2\partial_{x_1} C(tb, tb)b$$

we can solve

$$\begin{aligned} u(b) &= \int_0^1 \langle b, \partial_{x_1} C(tb, tb)^T \rangle dt = \int_0^1 \frac{1}{2}\frac{d}{dt} C(tb, tb) dt \\ &= \frac{1}{2} C(tb, tb)|_{t=0}^1 = \frac{C(b, b) - C(0, 0)}{2}. \end{aligned}$$

Step 4. The final step is to prove that

$$\begin{aligned} &\forall b : b \in \arg\max_a \{u(a) - C(a, b)\} \\ \Leftrightarrow \forall a, b : &\frac{C(b, b) - C(0, 0)}{2} - C(b, b) \geq \frac{C(a, a) - C(0, 0)}{2} - C(a, b) \\ \Leftrightarrow \forall a, b : &C(a, b) - \frac{C(a, a) + C(b, b)}{2} \geq 0 \end{aligned} \tag{5.16}$$

Now recall that in a generator set model, "max-prob" uses $-\log(\text{pdf}[x_1, x_2])$ as cost function, where x_1 and x_2 are generated from one same point $x \in P$ (inlier or outlier). Then this equation is the non-negativity condition on normalized cost function #1 (Section 4.1.3), since:

$$C(a, b) - \frac{C(a, a) + C(b, b)}{2} = -\log\left(\frac{h(x_1, x_2)}{\sqrt{h(x_1, x_1)h(x_2, x_2)}}\right) \tag{5.17}$$

or the same replacing h with $\tilde{h}$ and H with $\tilde{H}$ in the case with outliers (Section 4.3).

We can show that this is valid for any distribution such that $\text{pdf}[x_1|x]$ and $\text{pdf}[x_2|x]$ are the same distributions. This is valid for the symmetric outlier model described in Section 3.4: In this case $\text{pdf}[x_1|x] = \text{pdf}[x_1|x, \text{inlier}]P[\text{inlier}] + \text{pdf}[x_1|x, \text{outlier}]P[\text{outlier}] = (1 - q')p_y(x_1 - x) + q'\{p * p_y\}(x_1)$.

[14] Although we can show that the Hessian is also negative semidefinite, this will not help us solve step 4 this time, so we will omit the proof. We will solve step 4 with a different method this time.

Let $\eta(x_1, x) = \text{pdf}[x_1|x]$. Then,

$$h(x_1, x_2) = \text{pdf}[x_1, x_2] = \int_{\mathbb{R}^n} \eta(x_1, x)\eta(x_2, x)p(x)dx. \tag{5.18}$$

Combining equations 5.16, 5.17 and 5.18, we obtain:

$$-\log\left(\frac{\left(\int_{\mathbb{R}^n} p(x)\eta(a,x)\eta(b,x)dx\right)}{\sqrt{\left(\int_{\mathbb{R}^n} p(x)\eta(a,x)^2dx\right)\left(\int_{\mathbb{R}^n} p(x)\eta(b,x)^2dx\right)}}\right) \geq 0 \Leftrightarrow$$

$$\frac{\left(\int_{\mathbb{R}^n} p(x)\eta(a,x)^2dx\right)\left(\int_{\mathbb{R}^n} p(x)\eta(b,x)^2dx\right)}{\left(\int_{\mathbb{R}^n} p(x)\eta(a,x)\eta(b,x)dx\right)^2} \geq 1 \Leftrightarrow$$

$$\frac{\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(x)p(y)\eta(a,x)^2\eta(b,y)^2dxdy}{\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(x)p(y)\eta(a,x)\eta(b,x)\eta(a,y)\eta(b,y)dxdy} \geq 1$$

In the numerator we linked x to a and y to b, but the opposite would also be valid. Taking the mean between the two cases, we get:

$$\frac{\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(x)p(y)\frac{\eta(a,x)^2\eta(b,y)^2+\eta(a,y)^2\eta(b,x)^2}{2}dxdy}{\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(x)p(y)\eta(a,x)\eta(b,x)\eta(a,y)\eta(b,y)dxdy} \geq 1$$

It is sufficient to show that:

$$\frac{\eta(a,x)^2\eta(b,y)^2 + \eta(a,y)^2\eta(b,x)^2}{2} \geq \eta(a,x)\eta(b,x)\eta(a,y)\eta(b,y)$$

$$\Leftrightarrow \frac{\eta(a,x)^2\eta(b,y)^2 + \eta(a,y)^2\eta(b,x)^2}{2} - \eta(a,x)\eta(b,x)\eta(a,y)\eta(b,y) \geq 0$$

$$\Leftrightarrow \frac{\left(\eta(a,x)\eta(b,y) - \eta(a,y)\eta(b,x)\right)^2}{2} \geq 0$$

which is clear to be always satisfied.

It is then proved that $x_2^*(x_1) = x_1$ solves the variational problem with a generator set, for any distribution of P or the noise. We also proved that normalized cost function #1 is always non-negative.

Curiously, the result $x_2^*(x_1) = x_1$ applies to any non-negative cost function satisfying $\forall x : C(x, x) = 0$. This means that, in the generator set model, regardless of the distribution of P, one may choose for example L^2 or L^1 distance as cost and the expected hit count will converge to the same value when $N \to \infty$ as if one had chosen the correct cost function, i.e. the one based on the joint probability.

5.3 Computing the expected hit count

Now that we have computed the $x_2^*(x_1)$ function for multiple models and distributions, we are able to calculate the expected hit count as $N \to \infty$ for these

cases.

5.3.1 Gaussian case

Direct Model

In the isotropic Gaussian case, in the direct model, we obtain:

$$
\begin{aligned}
\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] &= \int_{\mathbb{R}^n} \frac{h(x_1, x_2^*(x_1))}{p_2(x_2^*(x_1))} dx_1 \\
&= \int_{\mathbb{R}^n} \frac{g_\sigma(x_1) g_\epsilon(x_2^*(x_1) - x_1)}{g_{\sqrt{\sigma^2+\epsilon^2}}(x_2^*(x_1))} dx_1 \\
&= \int_{\mathbb{R}^n} \frac{g_\sigma(x_1) g_\epsilon(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} x_1 - x_1)}{g_{\sqrt{\sigma^2+\epsilon^2}}(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} x_1)} dx_1 \\
&= \left(\frac{\sqrt{2\pi(\sigma^2+\epsilon^2)}}{\sqrt{2\pi\sigma^2}\sqrt{2\pi\epsilon^2}}\right)^n \int_{\mathbb{R}^n} \frac{e^{-||x_1||^2/(2\sigma^2)} \cdot e^{-\frac{(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma}-1)^2||x_1||^2}{2\epsilon^2}}}{\exp\left(-\frac{1}{2}\left(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} x\right)^2 / (\sigma^2+\epsilon^2)\right)} dx_1 \\
&= \left(\frac{\sqrt{2\pi(\sigma^2+\epsilon^2)}}{\sqrt{2\pi\sigma^2}\sqrt{2\pi\epsilon^2}}\right)^n \int_{\mathbb{R}^n} \frac{e^{-||x_1||^2/(2\sigma^2)} \cdot e^{-\frac{(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma}-1)^2||x_1||^2}{2\epsilon^2}}}{e^{-\frac{(\frac{x}{\sigma})^2}{2}}} dx_1 \\
&= \left(\frac{\sqrt{2\pi(\sigma^2+\epsilon^2)}}{\sqrt{2\pi\sigma^2}\sqrt{2\pi\epsilon^2}}\right)^n \int_{\mathbb{R}^n} e^{-\frac{(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma}-1)^2||x_1||^2}{2\epsilon^2}} dx_1 \\
&= \left(\frac{\sqrt{2\pi(\sigma^2+\epsilon^2)}}{\sqrt{2\pi\sigma^2}\sqrt{2\pi\epsilon^2}} \cdot \sqrt{2\pi} \frac{\epsilon}{\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} - 1}\right)^n \\
&= \left(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} \cdot \frac{1}{\frac{\sqrt{\sigma^2+\epsilon^2}}{\sigma} - 1}\right)^n \\
&= \left(\frac{1}{1 - \frac{\sigma}{\sqrt{\sigma^2+\epsilon^2}}}\right)^n \\
&= \left(\frac{1}{1 - \frac{1}{\sqrt{1+\frac{\epsilon^2}{\sigma^2}}}}\right)^n
\end{aligned}
$$

Analyzing the formula above we note that:

- As $\sigma \to 0$, we have $E[\#\text{hits}_{\text{max-prob}}] = 1$, which is the same result as when there is infinite noise with fixed N;
- $E[\#\text{hits}_{\text{max-prob}}]$ increases as the noise ratio ϵ/σ falls;
- $E[\#\text{hits}_{\text{max-prob}}]$ increases with the number of dimensions.

Therefore the formula agrees with the theoretical results found in the beginning of this chapter.

Generator set model

Analogously, in the generator set model, we obtain:

$$\begin{aligned}
\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] &= \int_{\mathbb{R}^n} \frac{h(x_1, x_2^*(x_1))}{p_2(x_2^*(x_1))} dx_1 \\
&= \int_{\mathbb{R}^n} \frac{g_{\sqrt{\sigma^2+\epsilon^2/2}}\left(\frac{x_1+x_2^*(x_1)}{2}\right) g_{\sqrt{2}\epsilon}(x_1 - x_2^*(x_1))}{g_{\sqrt{\sigma^2+\epsilon^2}}(x_2^*(x_1))} dx_1 \\
&= \int_{\mathbb{R}^n} \frac{g_{\sqrt{\sigma^2+\epsilon^2/2}}(x_1) g_{\sqrt{2}\epsilon}(0)}{g_{\sqrt{\sigma^2+\epsilon^2}}(x_1)} dx_1 \\
&= \left(\frac{\sigma^2+\epsilon^2}{2\pi \cdot 2\epsilon^2(\sigma^2+\epsilon^2/2)}\right)^{\frac{n}{2}} \int_{\mathbb{R}^n} \frac{e^{-\frac{1}{2}||x_1||^2/(\sigma^2+\epsilon^2/2)}}{e^{-\frac{1}{2}||x_1||^2/(\sigma^2+\epsilon^2)}} dx_1 \\
&= \left(\frac{\sigma^2+\epsilon^2}{2\pi \cdot 2\epsilon^2(\sigma^2+\epsilon^2/2)}\right)^{\frac{n}{2}} \int_{\mathbb{R}^n} e^{-\frac{1}{2}\left(\frac{1}{\sigma^2+\epsilon^2/2} - \frac{1}{\sigma^2+\epsilon^2}\right)||x_1||^2} dx_1 \\
&= \left(\frac{\sigma^2+\epsilon^2}{2\epsilon^2(\sigma^2+\epsilon^2/2)\left(\frac{1}{\sigma^2+\epsilon^2/2} - \frac{1}{\sigma^2+\epsilon^2}\right)}\right)^{\frac{n}{2}} \\
&= \left(\frac{\sigma^2+\epsilon^2}{2\epsilon^2(\sigma^2+\epsilon^2/2)\left(\frac{\epsilon^2/2}{(\sigma^2+\epsilon^2/2)(\sigma^2+\epsilon^2)}\right)}\right)^{\frac{n}{2}} \\
&= \left(\frac{(\sigma^2+\epsilon^2)^2}{\epsilon^4}\right)^{\frac{n}{2}} \\
&= \left(1+\frac{\sigma^2}{\epsilon^2}\right)^n
\end{aligned}$$

Note that this expression satisfies the same properties mentioned for the formula for the direct case. Also, interestingly, the generator set model case is approximately equal to the direct model case with $\sqrt{2}$ times as much noise: Note that

$$\left(\frac{1}{1-\frac{1}{\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}}}\right)^n = \left(\frac{\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}}{\sqrt{1+\frac{2\epsilon^2}{\sigma^2}}-1}\right)^n = \left(\frac{\left(\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}+1\right)\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}}{1+2\frac{\epsilon^2}{\sigma^2}-1}\right)^n =$$

$$\left(\frac{1+2\frac{\epsilon^2}{\sigma^2}+\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}}{2\epsilon^2/\sigma^2}\right)^n = \left(1+\frac{\sigma^2}{2\epsilon^2}+\frac{\sigma^2}{2\epsilon^2}\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}\right)^n =$$

$$\left(1+\frac{\sigma^2}{2\epsilon^2}+\frac{\sigma^2}{2\epsilon^2}\left(1+\frac{2\frac{\epsilon^2}{\sigma^2}}{\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}+1}\right)\right)^n = \left(1+\frac{\sigma^2}{\epsilon^2}+\frac{1}{\sqrt{1+2\frac{\epsilon^2}{\sigma^2}}+1}\right)^n$$

$$= \left(1 + \frac{\sigma^2}{\epsilon^2} + O(1)\right)^n$$

5.3.2 Exponential case

Exponential distribution model

Suppose now that the points in the generator set P have an exponential distribution, and that noise has an isotropic Gaussian distribution of parameter ϵ.

For an n-dimensional exponential distribution, it is reasonable to let $p(x) \propto e^{-\lambda||x||}$. In this case, we have:

$$\int_{\mathbb{R}^n} e^{-\lambda||x||} dx = \int_0^\infty e^{-\lambda r} A_n r^{n-1} dr = \frac{A_n(n-1)!}{\lambda^n}$$

where A_n is the $(n-1)$-dimensional hyper-area of the boundary of an n-dimensional hyper-sphere of radius 1 (see Appendix C).

Therefore our distribution is:

$$p(x) = \frac{\lambda^n e^{-\lambda||x||}}{A_n(n-1)!}$$

Note that when $n = 1$, we obtain $p(x) = \frac{\lambda}{2} e^{-\lambda|x|}$, and not the standard exponential distribution $\text{pdf}[t] = \lambda e^{-\lambda t}$, since the latter only considers positive values for t. This distribution is also known as the Laplace Distribution (when $n = 1$).

In fact, for the theoretical properties of matching that we are interested in deriving, the important characteristic of our exponential distribution model is that $p(x) = \Theta(e^{-\lambda||x||})$ as $||x|| \to \infty$, so with similar distributions we would get the same theoretical results.

Expected hit count

With isotropic Gaussian noise and a generator set model, any distribution has $x_2^*(x_1) = x_1$, so the expected hit count is:

$$\begin{aligned}
\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] &= \int_{\mathbb{R}^n} \frac{h(x_1, x_1)}{p_2(x_1)} dx_1 \\
&= \int_{\mathbb{R}^n} \frac{p_m(x_1) g_{\sqrt{2}\epsilon}(0)}{p_2(x_1)} dx_1 \\
&= \frac{1}{(4\pi\epsilon^2)^{n/2}} \int_{\mathbb{R}^n} \frac{\left\{p * g_{\epsilon/\sqrt{2}}\right\}(x_1)}{\left\{p * g_\epsilon\right\}(x_1)} dx_1
\end{aligned}$$

A necessary condition for this integral to converge is that:

$$\lim_{||x_1||\to\infty} Z(x_1) = 0$$
$$\text{where } Z(x_1) = \frac{\left\{p * g_{\epsilon/\sqrt{2}}\right\}(x_1)}{\left\{p * g_{\epsilon}\right\}(x_1)}$$

otherwise, if $\lim_{||x_1||\to\infty} Z(x_1) > 0$, the integral is infinite and we have an infinite expected hit count.

If $p(x)$ is Gaussian we know that $\lim_{||x_1||\to\infty} Z(x_1) = 0$: If $p(x) = g_\sigma(x)$, then $Z(x_1) = \frac{g_{\sqrt{\sigma^2+\epsilon^2/2}}(x_1)}{g_{\sqrt{\sigma^2+\epsilon^2}}(x_1)} = \left(\frac{\sigma^2+\epsilon^2}{\sigma^2+\epsilon^2/2}\right)^{n/2} \exp\left(-\frac{1}{2}\frac{\epsilon^2/2||x_1||^2}{(\sigma^2+\epsilon^2)(\sigma^2+\epsilon^2/2)}\right)$, which tends to zero as $||x_1|| \to \infty$.

However, if $p(x)$ is the exponential distribution described in Section 5.3.2, we have another result.

Let us rewrite:

$$\lim_{||x_1||\to\infty} \frac{\left\{p * g_{\epsilon/\sqrt{2}}\right\}(x_1)}{\left\{p * g_{\epsilon}\right\}(x_1)} = \frac{\lim_{||x_1||\to\infty}\left\{p * g_{\epsilon/\sqrt{2}}\right\}(x_1)/p(x_1)}{\lim_{||x_1||\to\infty}\left\{p * g_{\epsilon}\right\}(x_1)/p(x_1)}$$

Let $x = x_1 + a + b$, for two vectors a, b, where $a \parallel x_1$ and $b \perp x_1$. Then,

$$\lim_{||x_1||\to\infty} \left\{p * g_{\epsilon}\right\}(x_1)/p(x_1) =$$

$$\lim_{||x_1||\to\infty} \frac{\int_{\mathbb{R}^n} \frac{\lambda^n}{A_n(n-1)!} e^{-\lambda||x||} \cdot \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} dx}{\frac{\lambda^n}{A_n(n-1)!} e^{-\lambda||x_1||}} =$$

$$\lim_{||x_1||\to\infty} \frac{\int_{\mathbb{R}^n} e^{-\lambda||x||} \cdot \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} dx}{e^{-\lambda||x_1||}} =$$

$$\lim_{||x_1||\to\infty} \frac{\int_{a\parallel x_1} \int_{b\perp x_1} e^{-\lambda||x_1+a+b||} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} dadb}{e^{-\lambda||x_1||}} =$$

$$\int_{a\parallel x_1} \int_{b\perp x_1} e^{-\lambda \lim_{||x_1||\to\infty}(||x_1+a+b||-||x_1||)} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} dadb \tag{5.19}$$

Let us take a closer look at $\lim_{||x_1||\to\infty}(||x_1 + a + b|| - ||x_1||)$.

$$\lim_{||x_1||\to\infty}(||x_1 + a + b|| - ||x_1||) = \lim_{||x_1||\to\infty} \frac{(||x_1 + a + b|| - ||x_1||)(||x_1 + a + b|| + ||x_1||)}{||x_1 + a + b|| + ||x_1||}$$

$$= \lim_{||x_1||\to\infty} \frac{||x_1 + a + b||^2 - ||x_1||^2}{||x_1 + a + b|| + ||x_1||}$$

$$= \lim_{||x_1||\to\infty} \frac{||a+b||^2 + 2\langle x_1, a+b\rangle}{||x_1+a+b|| + ||x_1||}$$

$$= \lim_{||x_1||\to\infty} \frac{||a+b||^2 + 2\langle x_1, a\rangle}{||x_1+a+b|| + ||x_1||} = \left\langle \frac{x_1}{||x_1||}, a \right\rangle$$

Substituting in Equation 5.19, we obtain:

$$\int_{a\| x_1} \int_{b\perp x_1} e^{-\lambda\left\langle \frac{x_1}{||x_1||}, a\right\rangle} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}} da db$$

$$= \int_{a\| x_1} e^{-\lambda\left\langle \frac{x_1}{||x_1||}, a\right\rangle} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}}}{\sqrt{2\pi\epsilon^2}} da$$

$$= \int_{a\| x_1} \frac{e^{-\frac{1}{2}\frac{||a+\lambda\epsilon^2\frac{x_1}{||x_1||}||^2}{\epsilon^2} + \frac{\lambda^2\epsilon^2}{2}}}{\sqrt{2\pi\epsilon^2}} da$$

$$= e^{\lambda^2\epsilon^2/2}$$

In fact, a more cautious proof would require to show that bringing the limit to inside the integral as we did in Equation 5.19 is allowed in this case; we skip this step here. Intuitively, it is allowed because the values of (a, b) where the limit does not apply for a given x_1 are highly attenuated by the Gaussian factor $e^{-||a||^2/2\epsilon^2}$, so they can be disregarded. For instance, if $a + b = -x_1$, then $e^{-\lambda(||x_1+a+b||-||x_1||)} = e^{\lambda||x_1||}$ grows exponentially with x_1, but the Gaussian factor is $e^{-a^2/2\epsilon^2} = e^{-||x_1||^2/2\epsilon^2}$, which decreases much faster than the former.

This result shows nevertheless that, in the exponential case, $\lim_{x_1\to\infty} Z(x_1) = e^{\lambda^2(\epsilon/\sqrt{2})^2/2}/e^{\lambda^2\epsilon^2/2} = e^{-\lambda^2\epsilon^2/4} > 0$, which means that the expected hit count goes to infinity. In other words, as $N \to \infty$, $E[\#\text{hits}_{\text{max-prob}}] = \omega(1)$ for the exponential distribution.

5.3.3 Power law case

Power law distribution Model

Consider now a power law distribution of the points of P, and once again Gaussian isotropic noise.

For a power law distribution, similarly to the exponential distribution, we would like some $p(x)$ such that $p(x) = \Theta(||x||^{-\alpha})$. A convenient distribution with this property is the one that satisfies:

$$P[||x|| > t] = (m/t)^{\alpha-n},\ t > m$$

$$P[||x|| < m] = 0$$

for a scale parameter m. Also note that this model requires $\alpha > n$. Deriving the CDF with respect to $||x||$, we get

$$\text{pdf}[||x||] = (n-\alpha)\left(\frac{m^{\alpha-n}}{||x||^{\alpha-n+1}}\right)$$

And therefore:

$$p(x) = \frac{\text{pdf}[||x||]}{A_n||x||^{n-1}} = \frac{n-\alpha}{A_n}\frac{m^{\alpha-n}}{||x||^{\alpha}}$$

Expected hit count

Analogously to the exponential distribution, we have to compute $\lim_{x_1\to\infty} Z(x_1)$ for the power law distribution.

$$\lim_{||x_1||\to\infty} \{p * g_\epsilon\}(x_1)/p(x_1) =$$

$$\lim_{||x_1||\to\infty} \frac{\int_{||x||>m} \frac{(n-\alpha)m^{\alpha-n}}{A_n}||x||^{-\alpha} \cdot \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}dx}{\frac{(n-\alpha)m^{\alpha-n}}{A_n}||x_1||^{-\alpha}} =$$

$$\lim_{||x_1||\to\infty} \int_{||x||>m} \left(\frac{||x_1||}{||x||}\right)^{\alpha} \cdot \frac{e^{-\frac{1}{2}\frac{||x_1-x||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}dx =$$

$$\lim_{||x_1||\to\infty} \iint_{a\parallel x_1,\, b\perp x_1,\, ||x_1+a+b||>m} \left(\frac{||x_1||}{||x_1+a+b||}\right)^{\alpha} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}dadb =$$

$$\iint_{a\parallel x_1,\, b\perp x_1,\, ||x_1+a+b||>m} \left(\lim_{||x_1||\to\infty} \frac{||x_1||}{||x_1+a+b||}\right)^{\alpha} \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}dadb =$$

$$\int_{a\parallel x_1}\int_{b\perp x_1} 1 \cdot \frac{e^{-\frac{1}{2}\frac{||a||^2}{\epsilon^2}} e^{-\frac{1}{2}\frac{||b||^2}{\epsilon^2}}}{(2\pi\epsilon^2)^{n/2}}dadb = 1$$

The same issue with incorporating the limit to the integral happens here, but once again, the $(||x_1||/||x_1+a+b||)^{\alpha}$ term, which grows according to a power law, is attenuated by the Gaussianly decreasing factor $e^{-a^2/2\epsilon^2}$, so that the region where $a+b \sim -x_1$ can be disregarded. We skip the rigorous proof here.

Finally, we obtain that $\lim_{||x_1||\to\infty} Z(x_1) = 1/1 = 1 > 0$, which shows that the power law distribution of the generator set will also have an infinite number of correct matches as $N \to \infty$.

5.4 Expected hit count with outliers

We have seen that, in the generator set model, $x_2^*(x_1) = x_1$ is solution to the variational problem regardless of whether there are outliers or not, for any distribution. In the direct model, we have seen that $x_2^*(x_1)$ is the same for both cases, with or without outliers, when distributions are Gaussian.

So for these cases, the only difference in the derivation of the expected hit count is in that $P[\text{hit}] = (1-q)/N$ instead of $1/N$. We obtain then

$$P[\text{hit}|x_1, x_2] = \frac{\text{pdf}[x_1, x_2|\text{hit}]P[\text{hit}]}{\text{pdf}[x_1, x_2|\text{hit}]P[\text{hit}] + \text{pdf}[x_1, x_2|\neg\text{hit}]P[\neg\text{hit}]}$$

$$= \frac{(1-q)/N}{(1-q)/N + \frac{N-(1-q)}{N} \cdot \frac{p_1(x_1)p_2(x_2)}{h(x_1,x_2)}}$$

giving

$$\lim_{N\to\infty} E[\#\text{hits}_{\text{max-prob}}] = N \int_{\mathbb{R}^n} \frac{(1-q)/N}{(1-q)/N + \frac{N-(1-q)}{N} \cdot \frac{p_1(x_1)p_2(x_2^*(x_1))}{h(x_1,x_2^*(x_1))}} \cdot p_1(x_1)dx_1$$

$$= (1-q) \int_{\mathbb{R}^n} \frac{h(x_1, x_2^*(x_1))}{p_2(x_2^*(x_1))} dx_1$$

Therefore, the results obtained are the same but multiplied by $(1-q)$ for all the cases we have seen.

5.5 Experiments

5.5.1 Variational Problem

The purpose of this experiment is to corroborate the solution of the variational problem for Gaussian distributions and squared Euclidean distance cost: $x_2^*(x_1) = \Sigma_2 Q \Sigma_1^{-1} x_1$. We randomly generate two sets of points P_1 and P_2 independently (i.e. not following any model of Chapter 3, but simply independently[15]) following Gaussian distributions, apply minimum bipartite matching using squared Euclidean distance cost, and measure the average value of $||x_2^*(x_1) - x_2||^2$ among all matched pairs (x_1, x_2), where $x_2^*(x_1)$ is the solution to the variational problem. We repeat this measurement for increasing N and analyze how this metric changes: If x_2^* is correct, the metric should converge to zero, otherwise, it converges to a constant greater than zero.

[15]Note that the results on the variational problem depend only on the cost function $C(x_1, x_2)$ and on the prior probabilities $\text{pdf}[x_1]$ and $\text{pdf}[x_2]$, so they are indifferent to whether P_1 and P_2 were generated independently or according to a model such as the direct model or the generator set model.

We start computing this with $n = 1$ and $\sigma_1 = \sigma_2 = 1$ (Figure 5.1(a)) and $2\sigma_1 = 1$ and $\sigma_2 = 2$ (Figure 5.1(b)). Because $n = 1$, we can compute this for very high N using the sorting solution ($O(N \log N)$) described in Section 4.1.5. In the first case, we observe that $||x_1 - x_2||^2$ is decreasing apparently according to a power law of N, suggesting that x_2^* is correct. In the second case, we compare the behavior of $||x_2^*(x_1) - x_2||^2$ for $x_2^*(x_1) = 2x_1$ (correct) and $x_2^*(x_1) = x_1$ (incorrect), and we can clearly see that the former goes to zero while the latter does not. In Figure 5.1(c,d) we repeat this for $n = 2$ and isotropic distributions, this time using the Hungarian algorithm ($O(N^3)$), and observe the same behavior. In Figure 5.1(e) we use anisotropic distributions of variance $\Sigma_1^2 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 4 \end{smallmatrix}\right]$ and $\Sigma_2^2 = \left[\begin{smallmatrix} 5 & 3 \\ 3 & 5 \end{smallmatrix}\right]$, which means $x_2^*(x_1) = Tx_1$ with $T = \left[\begin{smallmatrix} 2.03026 & 0.468521 \\ 0.468521 & 1.09322 \end{smallmatrix}\right]$. It is clear in the graph that in this case $||x_2^*(x_1) - x_2||^2$ approaches zero, as expected.

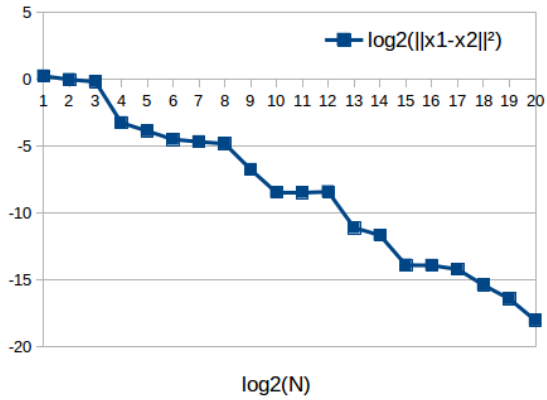
We also analyzed if the variational problem solution also applies to the Greedy #2 algorithm. We fixed $n = 1$ with Gaussian distributions and Euclidean distance as cost, and analyzed what happens when $\sigma_1 = \sigma_2 = 1$ (Figure 5.2(a)) and $\sigma_1 = 1$ and $\sigma_2 = 2$ (Figure 5.2(b)). In the former case, Greedy #2 converges to $x_2^*(x_1) = x_1$ just as minimum bipartite matching, but in the latter case, neither $x_2^*(x_1) = x_1$ nor $x_2^*(x_1) = 2x_1$ seem to be correct. This is an expected result, since the variational formulation seen in this chapter is based on minimum bipartite matching, not the greedy algorithms, whose behavior will be discussed in the next section.

5.5.2 Behavior of Greedy #2

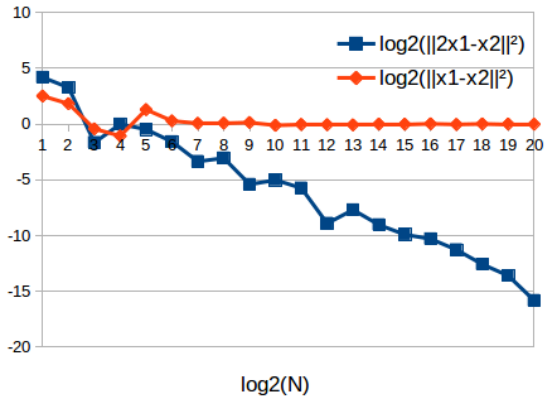
In order to understand what Greedy #2 is actually doing as $N \to \infty$, we did an experiment where we matched P_1 and P_2 (still independently generated) with $|P_1| = |P_2| = 10000$ and $n = 1$, using both "max-prob" and Greedy #2 methods, and we plotted the matched pairs in $\mathbb{R}^2$ (the point from P_1 in the x axis and P_2 in the y axis). We tested two scenarios: the case when the two distributions are equal ($\sigma_1 = \sigma_2 = 1$), and the case when they are different ($\sigma_1 = 1, \sigma_2 = 2$).

As the solution to the variational problem in each case is $x_2^*(x_1) = x_1$ and $x_2^*(x_1) = 2x_1$, "max-prob" results in a linear curve in both cases (Figure 5.3(a,b)). When $\sigma_1 = \sigma_2$, Greedy #2 will match first the points that are very close to each other, while in the final iterations, the matches are very distant to each other, so the plot looks like a straight line with noise (Figure 5.3(c)). When $\sigma_1 \neq \sigma_2$, however, Greedy #2 has a completely different behavior (Figure 5.3(d)).

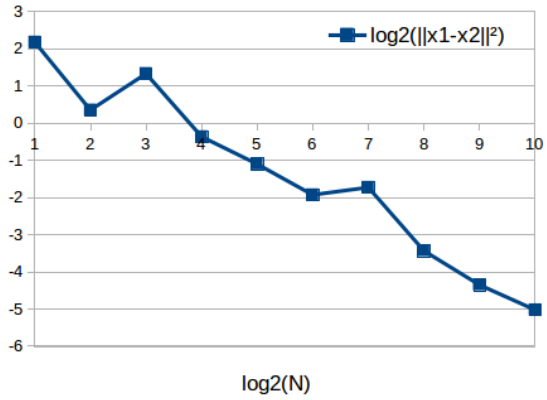
What is Greedy #2 actually doing in this case? Greedy #2 will start matching points that are very close to each other; however, matching must be one-to-one, so in the regions where P_1 and P_2 have different densities, at some point Greedy #2 will run out of points of either set and will have to match points very far from



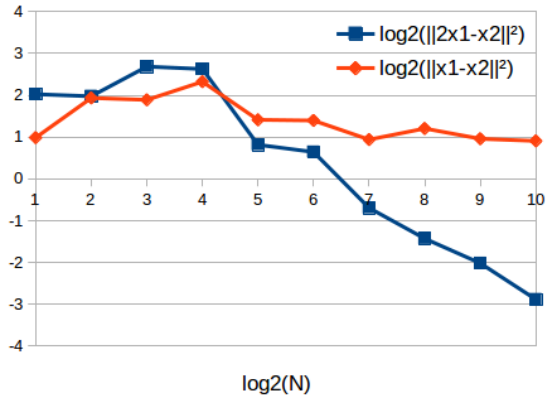
(a) $n = 1$, $\sigma_1 = \sigma_2 = 1$



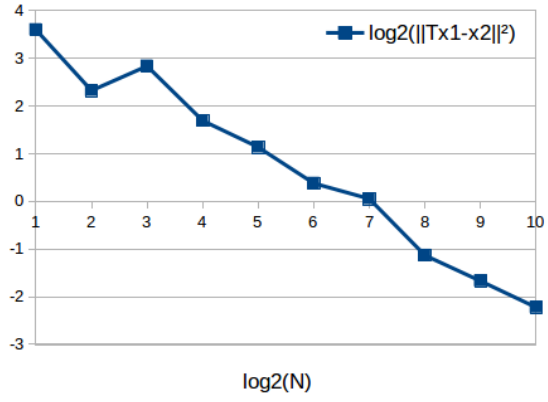
(b) $n = 1$, $\sigma_1 = 1$, $\sigma_2 = 2$



(c) $n = 2$, $\Sigma_1 = \Sigma_2 = I$

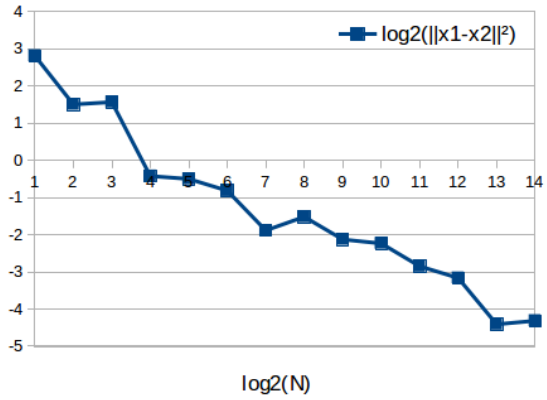


(d) $n = 2$, $\Sigma_1 = I$, $\Sigma_2 = 2I$

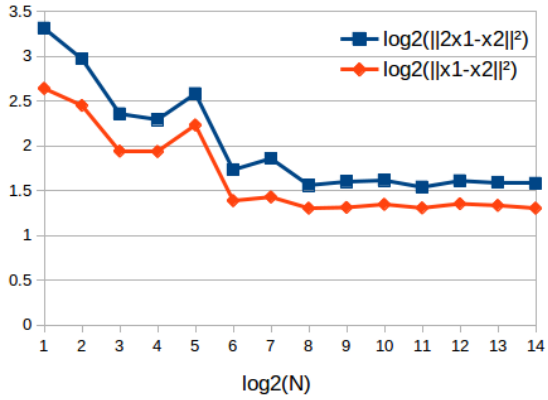


(e) $n = 2$, anisotropic

Figure 5.1: The convergence of $||x_2^*(x_1) - x_2||^2$ for different distributions and different functions for $x_2^*(x_1)$ (in log-log scale).

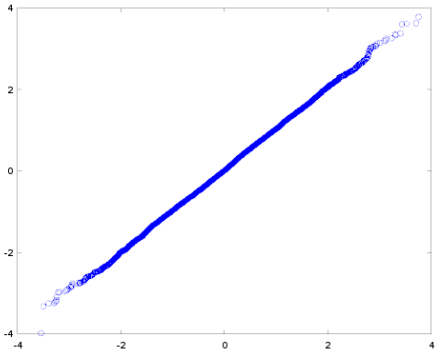


(a) $n = 1$, $\sigma_1 = \sigma_2 = 1$

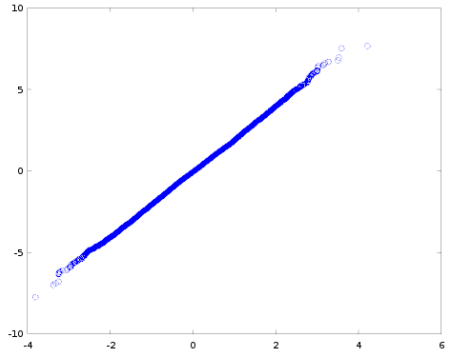


(b) $n = 1$, $\sigma_1 = 1$, $\sigma_2 = 2$

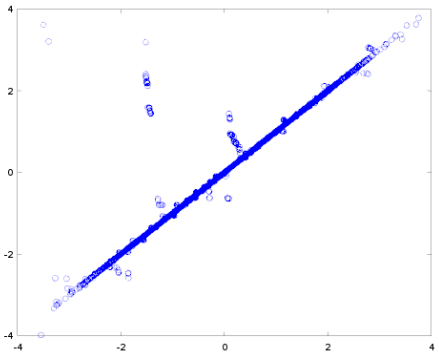
Figure 5.2: The convergence of $||x_2^*(x_1) - x_2||^2$ for Greedy #2.



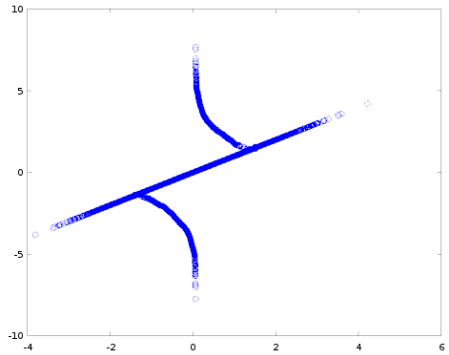
(a) "max-prob", $\sigma_1 = \sigma_2$



(b) "max-prob", $\sigma_1 \neq \sigma_2$



(c) Greedy #2, $\sigma_1 = \sigma_2$



(d) Greedy #2, $\sigma_1 \neq \sigma_2$

Figure 5.3: Plots of the matched pairs (x_1, x_2) with $n = 1$ and Gaussian distributions using different methods and different values of σ_1, σ_2. The x-axis shows the value of x_1 and the y-axis shows the value of x_2.

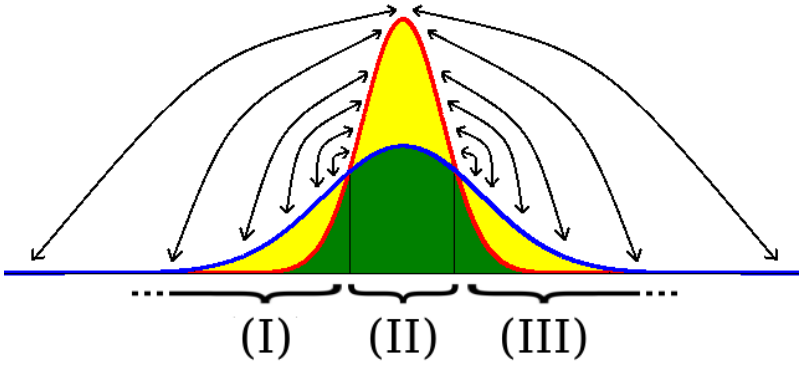


Figure 5.4: Illustration of the behavior of Greedy #2 with different distributions.

each other. Figure 5.4 illustrates its behavior. In the figure, Greedy #2 will start matching points from the green regions of the figure, and at some point regions (I) and (III) will run out of points of P_1 (red curve), while region (II) will run out of points of P_2 (blue curve). When this happens, Greedy #2 will have to match points between the yellow regions of the figure: the remaining points of P_1 in (II) and the remaining points of P_2 in (I) and (III). Because of the greedy nature of the algorithm, the remaining points will be matched in order of distance: It will start with the points near the boundaries between (I) and (II) and between (II) and (III) (represented with the short arrows in the figure), and in the end the points near the center of (II) and the extremes of (I) and (III) (represented with the long arrows). This explains the curves we see in Figure 5.3(d): the straight curve are the points matched in the first phase of the algorithm, i.e. before it ran out of points of P_1 or P_2 in the regions of different density, while the deviating curves are the points matched in the second phase of the algorithm, i.e. the matching of the remaining points.

5.5.3 Gaussian hit count

The goal of the experiments in this subsection is to confirm through simulations the expressions for the hit count when $N \to \infty$, for isotropic Gaussian distributions without outliers. We run "max-prob" for different values of N, n and ϵ/σ in the direct and generator set models and compare with the theoretical value predicted for $N \to \infty$.

We first analyze the direct model with $n = 1$, which means we can solve the problem in $O(N \log N)$ with the sorting solution. We ran "max-prob" for $\epsilon/\sigma \in \{.5, .75, 1, ..., 2\}$ and $N \in \{5, 50, 500, ..., 5 \cdot 10^6\}$, and took the average hit count for 100 samples (Figure 5.5(a,b)). The figures suggest that "max-prob" is converging to the theoretical value. We also compare this values with the hit count of Greedy

#2[16] (Figure 5.5(c) shows the values fixing $N = 5 \cdot 10^5$ and Figure 5.5(d) fixing $\epsilon/\sigma = 1$). It is clear in Figure 5.5(d) that Greedy #2 does not converge to the theoretical value, and has a lower hit count.

For low values of ϵ/σ, "max-prob" does not appear to converge to the theoretical value at first glance (Figure 5.5(e) shows the average hit count when $\epsilon/\sigma = .25$ and $n = 1$, using "max-prob" and Greedy #2 with 10 samples each, in the direct model). Our hypothesis is that it does converge, although very slowly. To support this hypothesis, our argument is that the integral of Equation 5.2 also converges very slowly as $N \to \infty$. Although we cannot compute this integral analytically for fixed N, we can estimate it using a Monte-Carlo method[17], which has $O(1)$ complexity with respect to N, meaning that we can compute it for extremely high N. We compare then the hit count of "max-prob" to this Monte-Carlo integral result, observe that they yield similar values[18], and also verify that the Monte-Carlo integral converges very slowly as $N \to \infty$ (Figure 5.5(f), where $n = 1$, $\epsilon/\sigma = .25$, with the direct model; "max-prob" uses 10 samples and Monte-Carlo uses 1000 samples).

In the generator set model, with $n = 1$, "max-prob" seems to converge correctly to the theoretical value (Figure 5.6(a); "max-prob" hit count averaged with 100 samples). The difference is that, because the greedy method solution also converges to the same mapping function $x_2^*(x_1)$, its hit count also converges to the same hit count as "max-prob" (Figure 5.6(b); fixed $\epsilon/\sigma = .75$, both algorithms were averaged with 100 samples).

When $n > 1$, we cannot run "max-prob" or Greedy #2 with a high number of samples because their $O(N \log N)$ solutions cannot be applied in this case, so the analysis is much poorer (See Figure 5.7(a) for $n = 2$ and Figure 5.7(b) for $n = 3$, using the direct model and 50 samples per case).

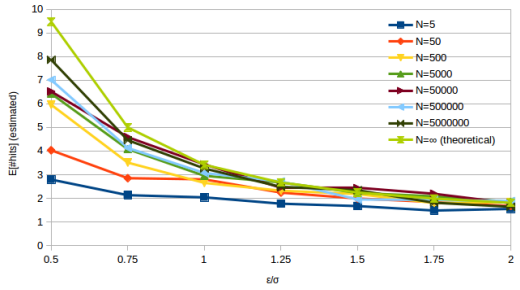
5.5.4 Exponential and power law hit count

The goal of this experiment is to verify that exponential and power law distributions have infinite hit count as $N \to \infty$. For that purpose we use the generator set model with $n = 1$ and run bipartite matching using squared Euclidean distance as cost (i.e. the sorting solution), since we do not have an analytical expression for the cost function for these distributions. However, both methods have the same solution as $N \to \infty$, because the solution of the variational problem is the same (as seen in

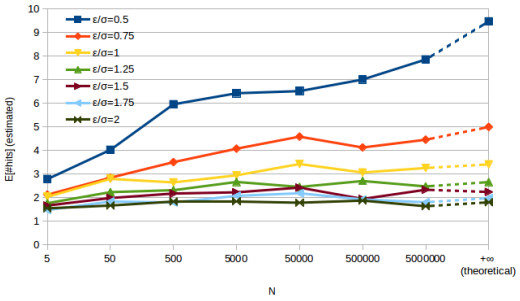
[16]Greedy #2 can also be computed in $O(N \log N)$ when $n = 1$ and Euclidean distance is used as cost function (see Appendix D).

[17]Our Monte-Carlo estimator samples x_1 with $\text{pdf}[x_1] \propto h(x_1, x_2^*(x_1))/p_2(x_2^*(x_1))$.

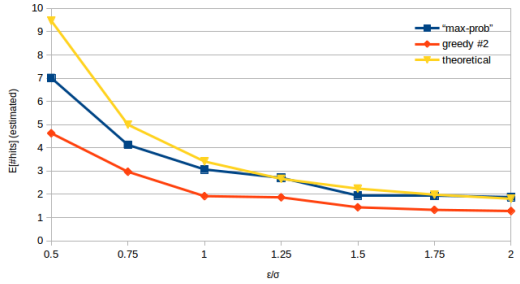
[18]Note that the integral of Equation 5.2 does not predict the hit count of "max-prob" for fixed N, as it already incorporate terms that are only valid for $N \to \infty$. We can only expect that they converge to the same value and have similar convergence rates.



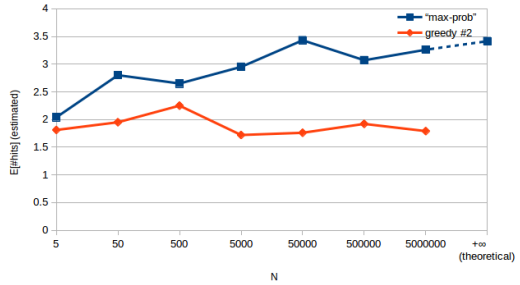
(a) Comparing "max-prob" with different values of N and ϵ/σ and the theoretical value for $N \to \infty$



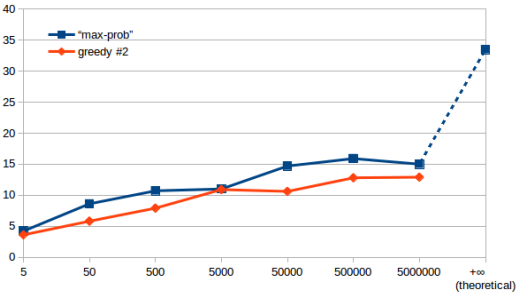
(b) Same as (a), plotted differently



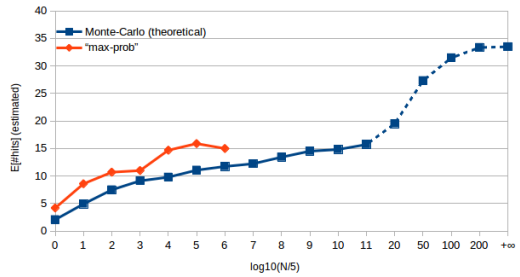
(c) Comparing "max-prob" and Greedy #2 for $N = 5 \cdot 10^5$ and the theoretical value for $N \to \infty$



(d) Comparing "max-prob", Greedy #2 and the theoretical value for $\epsilon/\sigma = 1$

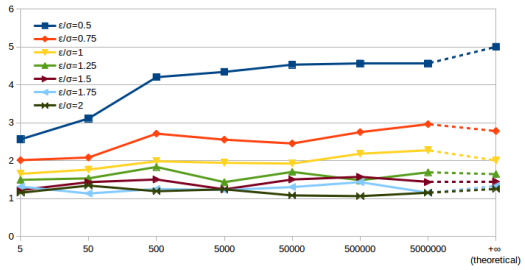


(e) Comparing "max-prob", Greedy #2 and the theoretical value for $\epsilon/\sigma = .25$

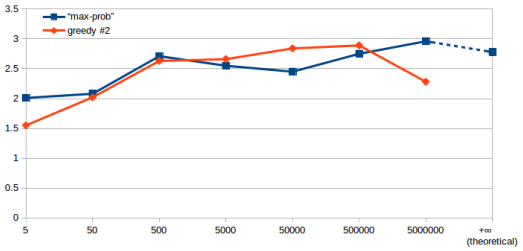


(f) Comparing "max-prob", Monte-Carlo integration (theoretical) for fixed N, and the theoretical formula for $N \to \infty$, with $\epsilon/\sigma = .25$. NOTE: The x-axis is not in a uniform scale.

Figure 5.5: Direct model comparisons, $n = 1$. In each chart, "$+\infty$ (theoretical)" refers to the (exact) theoretical value for $N \to \infty$, all others are numerically estimated.

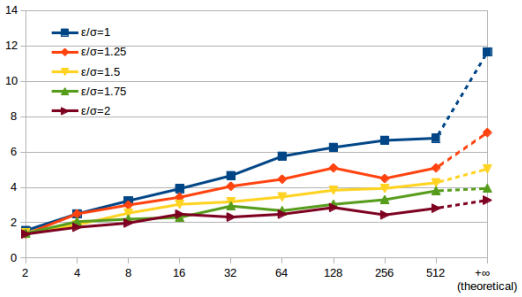


(a) Comparing "max-prob" with different values of N and ϵ/σ and the theoretical value for $N \to \infty$

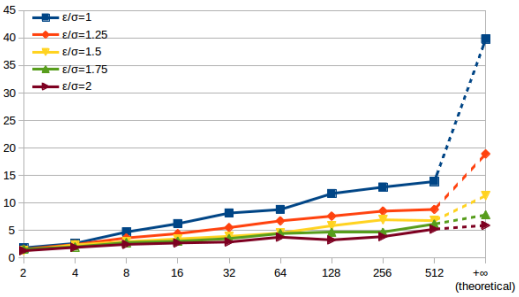


(b) Comparing "max-prob", Greedy #2 and the theoretical value for $\epsilon/\sigma = .75$

Figure 5.6: Generator set model comparisons, $n = 1$. In each chart, "$+\infty$ (theoretical)" refers to the (exact) theoretical value for $N \to \infty$, all others are numerically estimated.

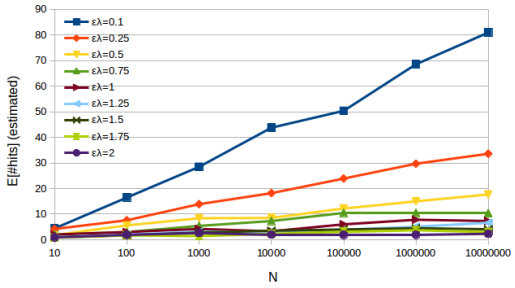


(a) $n = 2$

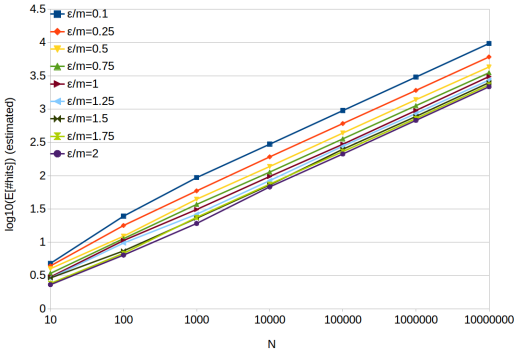


(b) $n = 3$

Figure 5.7: Direct model comparisons, $n > 1$ (including the theoretical value).



(a) Exponential distribution



(b) Power law distribution, $\alpha = 2$. NOTE: The y-axis is in logarithmic scale.

Figure 5.8: Hit count growth for exponential and power law distributions, $n = 1$.

Section 5.2.5, any non-negative cost function satisfying $\forall x : C(x, x) = 0$ solves the variational problem when points in P_1 and P_2 have the same prior distributions).

Figure 5.8(a) shows the average hit count for an exponential distribution with different values of $\epsilon\lambda$ and N while Figure 5.8(b) shows the hit count for a power law distribution with for different values of ϵ/m and N, and fixed $\alpha = 2$, both averaged over 10 samples. The former appears to show logarithmic growth ($E[\#\text{hits}] = \Theta(\log(N))$), while the latter shows a very clear power law growth in the hit count (Figure 5.8(b) suggests that $E[\#\text{hits}] = \Theta(\sqrt{N})$ in this case).

Chapter 6

Hit rate of Greedy #2

Rather than computing the exact expected number of correct matches as N grows to infinity $\lim_{N\to\infty} E[\#\text{hits}]$, as we did in the previous chapter, in this chapter we will pursue the expected hit count for some fixed N. In fact, we do not compute exact values; we rather derive bounds that describe an asymptotic behavior as N grows to infinity.

6.1 General idea

In this section we describe the basic reckoning we use to derive the bounds shown in this chapter.

Consider the generator set model with Gaussian noise and no outliers. Points in P_1 have a probability density of $p_1(x_1)$ that decreases as x_1 is farther from the origin. Therefore, points farther from the origin are more likely to be correctly matched, since the probability of appearing a neighboring point that could be mistaken with the correct match becomes gradually lower. This probability is related to the point density in the region, $Np_1(x_1)$. So let us assume that a point is correctly matched if the point density around the point is "sufficiently low":

$$P[\text{hit}] \approx P\left[Np_1(x_1) < c\right],$$

for some constant c, where x_1 is a random point in P_1, with distribution $\text{pdf}[x_1] = p_1(x_1)$. Theoretically the threshold should depend also on ϵ and n, but let us disregard this fact in this moment.

We can further approximate this relation by letting x_1 be distributed according to the distribution function of the generator set $p(x)$, obtaining:

$$P[\text{hit}] \approx P\left[p(x) < \frac{c}{N}\right]$$

Let us see some examples. If x is distributed according to a power law distribution, recall that we have:

$$P[||x|| > t] = (m/t)^{\alpha-n}$$

$$p(x) = \frac{n-\alpha}{A_n}\frac{m^{\alpha-n}}{||x||^\alpha}$$

Therefore, in this case,

$$p(x) < \frac{c}{N} \Leftrightarrow ||x|| > \left(\frac{A_n}{n-\alpha}\frac{1}{m^{\alpha-n}}\frac{c}{N}\right)^{-1/\alpha} = CN^{1/\alpha}$$

for a constant $C = \left(\frac{A_n}{n-\alpha}\frac{c}{m^{\alpha-n}}\right)^{-1/\alpha}$. We obtain then

$$P\left[p(x) < \frac{c}{N}\right] = \left(\frac{m}{CN^{1/\alpha}}\right)^{\alpha-n} = \left(\frac{m}{C}\right)^{\alpha-n} N^{n/\alpha-1}$$

suggesting that

$$P[\text{hit}] \approx C'N^{n/\alpha-1}$$

for constant C', and, because $E[\#\text{hits}] = NP[\text{hit}]$, that

$$E[\#\text{hits}] \approx C'N^{n/\alpha}.$$

On the other hand, for an exponential distribution (following the model from Section 5.3.2), we have:

$$p(x) = \frac{\lambda^n}{A_n(n-1)!}e^{-\lambda||x||}$$

$$P[||x|| > t] = e^{-\lambda t}\left(1 + \lambda t + \frac{(\lambda t)^2}{2!} + ... + \frac{(\lambda t)^{n-1}}{(n-1)!}\right)$$

In this case,

$$p(x) < \frac{c}{N} \Leftrightarrow ||x|| > \frac{\log(CN)}{\lambda}$$

for some constant C. Meanwhile,

$$P\left[||x|| > \frac{\log(CN)}{\lambda}\right] = e^{-\log(CN)}\left(1 + \log(CN) + \frac{(\log(CN))^2}{2!} + ... + \frac{(\log(CN))^{n-1}}{(n-1)!}\right)$$

$$\sim \frac{1}{CN}\frac{(\log N)^{n-1}}{(n-1)!} \text{ (as } N \to \infty).$$

Therefore for sufficiently large N we would have an expected hit count of

$$E[\#\text{hit}] = N.P[\text{hit}] \approx C'(\log N)^{n-1}$$

Naturally, the derivations above use a lot of approximations and are therefore very inaccurate, but as we will show in the next sections, these approximations can be rewritten as bounds so that we are able to obtain reliable results with respect to the hit rate.

6.2 Lower bound

Instead of working with approximations as in the previous section, in this section we will work with reliable bounds to describe the asymptotic behavior of the hit rate with respect to N.

The lower bounds are based on the Greedy #2 algorithm, and consequently also apply to "max-expect", since it is known to have a higher expected hit count than any other method. However, the bounds do not necessarily apply to the "max-prob" method, since we have no guarantee that it will have a higher hit count than Greedy #2.

Also, we will suppose that $N \to \infty$, while $\epsilon \to 0$ or $\epsilon = const.$, but not $\epsilon \to \infty$; and all the other parameters (such as n, σ (isotropic Gaussian case), λ (exponential case), m and α (power law case)) are constant. We are considering here only isotropic Gaussian noise and only the generator set model, although we expect that the direct model has most likely the same asymptotic behavior. To make the derivation simpler, we will assume for now that there are no outliers; the case with outliers is detailed in Section 6.4.

To simplify the notation, let $E[\#\text{hits}]$ and $P[\text{hit}]$ refer respectively to the hit count and rate of the Greedy #2 method with Euclidean distance as cost[1] (that is, an abuse of notation for $E[\#\text{hits}_{\text{greedy\#2}}]$ and $P[\text{hit}_{\text{greedy\#2}}]$, compared to the notation used in the previous chapter). We can say that $E[\#\text{hits}]$ is N times the probability of correctly matching a random point $x_1 \in P_1$, i.e.:

$$E[\#\text{hits}] = N \cdot P[\text{hit}] = N \int_{\mathbb{R}^n} P[\text{hit}|x_1] p_1(x_1) dx_1$$

Let then x_2 be the correct match of x_1 and x the corresponding point of the generator set. We can then write:

$$\begin{aligned} E[\#\text{hits}] &= N \iint_{\mathbb{R}^n \times \mathbb{R}^n} P[\text{hit}|x_1, x_2] \text{pdf}[x_1] \text{pdf}[x_2|x_1] dx_1 dx_2 \\ &= N \iint_{\mathbb{R}^n \times \mathbb{R}^n} P[\text{hit}|x_1, x_2] \text{pdf}[x_1, x_2] dx_1 dx_2 \end{aligned}$$

We know that Greedy #2 matches x_1 and x_2 with 100% probability if x_1 is the

[1] Also, $P[\text{hit}|x_1]$ denotes the probability of matching a given point x_1 correctly using Greedy #2, when all the other points are not given.

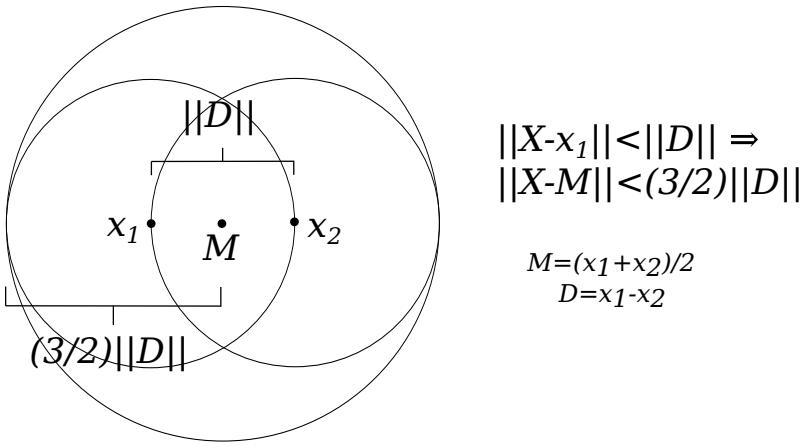


Figure 6.1: Illustration of the $\frac{3}{2}D$ radius sphere bound.

closest point to x_2 and vice-versa:

$$\begin{aligned} P[\text{hit}|x_1,x_2] \geq & P\left[\begin{array}{c}\left[\forall(x_2' \in P_2 \setminus \{x_2\}) : ||x_2' - x_1|| > ||x_2 - x_1||\right] \ \wedge ... \\ ... \left[\forall(x_1' \in P_1 \setminus \{x_1\}) : ||x_1' - x_2|| > ||x_2 - x_1||\right]\end{array} \middle| x_1, x_2\right] \\ = & P\left[\begin{array}{c}||\tilde{x}_2 - x_1|| > ||x_2 - x_1|| \ \wedge \\ ||\tilde{x}_1 - x_2|| > ||x_2 - x_1||\end{array} \middle| x_1, x_2\right]^{N-1} \end{aligned}$$

where $\tilde{x}_1$ and $\tilde{x}_2$ are random variables generated from a same point $\tilde{x}$ of the generator set, and "$\wedge$" denotes logical conjunction (the "and" operator). We can relax this condition to (see Figure 6.1 for an illustration of this step):

$$\begin{aligned} P[\text{hit}|x_1,x_2] \geq & P\left[\begin{array}{c}||\tilde{x}_2 - \frac{x_1+x_2}{2}|| > \frac{3}{2}||x_2 - x_1|| \ \wedge ... \\ ...||\tilde{x}_1 - \frac{x_1+x_2}{2}|| > \frac{3}{2}||x_2 - x_1||\end{array} \middle| x_1, x_2\right]^{N-1} \\ \triangleq & \ B\left(\frac{x_1+x_2}{2}, x_2 - x_1\right)^{N-1}. \end{aligned}$$

$B(M, D)$ can be interpreted as the probability of a pair $\tilde{x}_1, \tilde{x}_2$ being generated "far enough" from x_1 and x_2 so that matching is not hindered; given their mean $M = \frac{x_1+x_2}{2}$ and difference $D = x_1 - x_2$.

Recall that M and D are independent variables (as seen in Section 3.5), so that we can write now:

$$\begin{aligned} E[\#\text{hits}] \geq & N \iint_{\mathbb{R}^n \times \mathbb{R}^n} B(M,D)^{N-1} \text{pdf}[M]\text{pdf}[D] dM dD \\ = & N \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} B(M,D)^{N-1} \text{pdf}[D] dD \right) \text{pdf}[M] dM \end{aligned}$$

We can further restrain the integration domain to force $||D|| < \epsilon\bar{r}$, for some

constant $\bar{r}$:

$$E[\#\text{hits}] \geq N \int_{\mathbb{R}^n} \left(\int_{||D|| < \epsilon\bar{r}} B(M,D)^{N-1} \text{pdf}[D] dD \right) \text{pdf}[M] dM$$
$$= N \int_{\mathbb{R}^n} Q(M) \text{pdf}[M] dM$$

where:

$$Q(M) \triangleq \int_{||D|| < \epsilon\bar{r}} B(M,D)^{N-1} \text{pdf}[D] dD$$

$Q(M)$ can be interpreted as a lower bound of the probability of correctly matching x_1 and x_2, given their mean.

Using now Markov's inequality

$$\forall r : E[f(X)] \geq f(r) P[f(X) > f(r)],$$

we can use that for any $\bar{Q}$:

$$\int_{\mathbb{R}^n} Q(M) \text{pdf}[M] dM \geq \bar{Q} \cdot P[Q(M) \geq \bar{Q}]$$

$$\Rightarrow E[\#\text{hits}] \geq N\bar{Q} \cdot P[Q(M) \geq \bar{Q}].$$

In other words, we are bounding the expected hit rate to the probability of finding a point whose mean M implies a high probability (greater than $\bar{Q}$) of correctly matching, times this probability threshold ($\bar{Q}$). Naturally, this refers to points that are far enough from the origin. Now we need to solve $Q(M) \geq \bar{Q}$:

$$Q(M) \geq \bar{Q} \Leftrightarrow$$

$$\int_{||D|| < \epsilon\bar{r}} B(M,D)^{N-1} \text{pdf}[D] dD \geq \bar{Q} \Leftrightarrow$$

$$\int_{||D|| < \epsilon\bar{r}} \{1 - [1 - B(M,D)]\}^{N-1} \text{pdf}[D] dD \geq \bar{Q}$$

Because $(A \rightarrow B) \Rightarrow P[A] \leq P[B]$, we only need to find sufficient conditions for $Q(M) \geq \bar{Q}$. Using also $(1-p)^n \geq (1-np)$ for $p \in [0,1]$, we obtain:

$$Q(M) \geq \bar{Q} \Leftarrow \int_{||D|| < \epsilon\bar{r}} \{1 - (N-1)[1 - B(M,D)]\} \text{pdf}[D] dD \geq \bar{Q}$$

$$\Leftrightarrow \int_{||D|| < \epsilon\bar{r}} (N-1)(1 - B(M,D)) \text{pdf}[D] dD \leq P\left[||D|| < \epsilon\bar{r}\right] - \bar{Q}$$

$$\Leftrightarrow \int_{||D|| < \epsilon\bar{r}} (1 - B(M,D)) \text{pdf}[D] dD \leq \frac{P\left[||D|| < \epsilon\bar{r}\right] - \bar{Q}}{N-1}$$

Meanwhile, $B(M, D)$ is written:

$$B(M,D) = P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D|| \ \wedge \ ||\tilde{x}_2 - M|| > \frac{3}{2}||D|| \middle| D, M\right]$$

$$= \int_{\mathbb{R}^n} P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D|| \ \wedge \ ||\tilde{x}_2 - M|| > \frac{3}{2}||D|| \middle| \tilde{x}, D, M\right] \text{pdf}[\tilde{x}]d\tilde{x}$$

$$= \int_{\mathbb{R}^n} P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D|| \middle| \tilde{x}, D, M\right]^2 \text{pdf}[\tilde{x}]d\tilde{x}$$

$$= \int_{\mathbb{R}^n} \left(1 - P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D|| \middle| \tilde{x}, D, M\right]\right)^2 \text{pdf}[\tilde{x}]d\tilde{x}$$

$$\geq \int_{\mathbb{R}^n} \left(1 - 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D|| \middle| \tilde{x}, D, M\right]\right) \text{pdf}[\tilde{x}]d\tilde{x}$$

$$= 1 - 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D|| \middle| D, M\right].$$

Therefore:

$$Q(M) \geq \bar{Q} \Leftarrow$$

$$\int_{||D|| < \epsilon\bar{r}} 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D|| \middle| D, M\right] \text{pdf}[D]dD \leq \frac{P\left[||D|| < \epsilon\bar{r}\right] - \bar{Q}}{N-1}$$

$$\Leftarrow \int_{||D|| < \epsilon\bar{r}} 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}\epsilon\bar{r} \middle| D, M\right] \text{pdf}[D]dD \leq \frac{P\left[||D|| < \epsilon\bar{r}\right] - \bar{Q}}{N-1}$$

$$\Leftrightarrow 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}\epsilon\bar{r} \middle| M\right] P\left[||D|| < \epsilon\bar{r}\right] \leq \frac{P\left[||D|| < \epsilon\bar{r}\right] - \bar{Q}}{N-1}$$

$$\Leftrightarrow P\left[||\tilde{x}_1 - M|| < \frac{3}{2}\epsilon\bar{r} \middle| M\right] \leq \frac{1 - \frac{\bar{Q}}{P[||D|| < \epsilon\bar{r}]}}{2(N-1)}$$

Using the fact that the probability of $\tilde{x}_1$ being inside a given sphere is less than or equal to the volume of the sphere times the greatest probability density found for $\tilde{x}_1$ inside that sphere, we can write:

$$Q(M) \geq \bar{Q} \Leftarrow \frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \left\{\max_{||y-M|| < \frac{3}{2}\epsilon\bar{r}} p_1(y)\right\} \leq \frac{1 - \frac{\bar{Q}}{P[||D|| < \epsilon\bar{r}]}}{2(N-1)} \tag{6.1}$$

where $\frac{A_n}{n}$ is the hyper-volume of the hyper-sphere of radius 1 in $\mathbb{R}^n$ (See Appendix C).

6.2.1 Gaussian case

In an isotropic Gaussian model, Equation 6.1 is simplified if we further restrain

$||M|| > \frac{3}{2}\epsilon\bar{r}$, so that $\max_{||y-M||<\frac{3}{2}\epsilon\bar{r}} p_1(y) = p_1(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r})$, obtaining:

$$P[Q(M) \geq \bar{Q}] \geq P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \left\{\max_{||y-M||<\frac{3}{2}\epsilon\bar{r}} p_1(y)\right\} \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right)\right]$$

$$\geq P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n p_1\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right) \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right) \wedge ||M|| > \frac{3}{2}\epsilon\bar{r}\right]$$

$$= P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \frac{e^{-\frac{\left(||M|| - \frac{3}{2}\epsilon\bar{r}\right)^2}{2(\sigma^2+\epsilon^2)}}}{(2\pi(\sigma^2+\epsilon^2))^{n/2}} \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right) \wedge ||M|| > \frac{3}{2}\epsilon\bar{r}\right]$$

$$= P\left[||M|| \geq \frac{3}{2}\epsilon\bar{r} + \sqrt{2(\sigma^2+\epsilon^2)\log\left(\frac{C\epsilon^n(N-1)}{(\sigma^2+\epsilon^2)^{n/2}}\right)} \wedge ||M|| > \frac{3}{2}\epsilon\bar{r}\right] \tag{6.2}$$

Because we are interested in the asymptotic behavior only, we can use that:

$$\int_{||x||>r} \frac{e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}}{(2\pi\sigma^2)^{n/2}} dx \sim \frac{e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}}{(2\pi)^{n/2}} \cdot A_n (r/\sigma)^{n-2} \text{ (as } r \to \infty) \tag{6.3}$$

i.e.,

$$\lim_{r\to\infty} \frac{\int_{||x||>r} \frac{e^{-\frac{1}{2}\frac{||x||^2}{\sigma^2}}}{(2\pi\sigma^2)^{n/2}} dx}{\frac{e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}}{(2\pi)^{n/2}} \cdot A_n (r/\sigma)^{n-2}} =$$

$$\lim_{r\to\infty} \frac{-\frac{e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}}{(2\pi\sigma^2)^{n/2}} A_n r^{n-1}}{-\frac{r}{\sigma^2}\frac{e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}}{(2\pi)^{n/2}} \cdot A_n (r/\sigma)^{n-2} + (n-2)\frac{1}{\sigma}\frac{e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}}{(2\pi)^{n/2}} \cdot A_n (r/\sigma)^{n-3}} = 1$$

so that we can write:

$$E[\#\text{hits}] \geq N\bar{Q}P\left[||M|| \geq \frac{3}{2}\epsilon\bar{r} + \sqrt{2(\sigma^2+\epsilon^2)\log\left(\frac{C\epsilon^n(N-1)}{(\sigma^2+\epsilon^2)^{n/2}}\right)} \wedge ||M|| > \frac{3}{2}\epsilon\bar{r}\right]$$

$$\sim N\bar{Q}\frac{e^{-T^2/2}}{(2\pi(\sigma^2+\epsilon^2/2))^{n/2}} A_n T^{n-2} \tag{6.4}$$

$$\text{where } T = \frac{\frac{3}{2}\epsilon\bar{r} + \sqrt{2(\sigma^2+\epsilon^2)\log\left(\frac{C\epsilon^n(N-1)}{(\sigma^2+\epsilon^2)^{n/2}}\right)}}{\sqrt{\sigma^2+\epsilon^2/2}}$$

Equation 6.4 produces different results depending on the behavior of T.

If ϵ is constant, T depends only on N, so the equation reduces to

$$E[\#\text{hits}] = \Omega\left(NN^{-\frac{\sigma^2+\epsilon^2}{\sigma^2+\epsilon^2/2}}e^{-\frac{3}{2}\epsilon\bar{r}\sqrt{\frac{2(\sigma^2+\epsilon^2)}{\sigma^2+\epsilon^2/2}}\sqrt{\log N}}(\log N)^{\frac{n-2}{2}}\right)$$

$$= \Omega\left(N^{-\frac{\epsilon^2/2}{\sigma^2+\epsilon^2/2}}(\log N)^{\frac{n-2}{2}}e^{-\frac{3}{2}\epsilon\bar{r}\sqrt{\frac{2(\sigma^2+\epsilon^2)}{\sigma^2+\epsilon^2/2}}\sqrt{\log N}}\right)$$

Notice that both $N^{-\frac{\epsilon^2/2}{\sigma^2+\epsilon^2/2}}(\log N)^{\frac{n-2}{2}}$ and $e^{-\frac{3}{2}\epsilon\bar{r}\sqrt{\frac{2(\sigma^2+\epsilon^2)}{\sigma^2+\epsilon^2/2}}\sqrt{\log N}}$ tend to zero, therefore this is a looser bound that the one found in the last chapter for this case[2] ($E[\#\text{hits}] = \Omega(1)$, since $\lim_{N\to\infty} E[\#\text{hits}] = (1+\sigma^2/\epsilon^2)^n$).

If $\epsilon \to 0$ satisfying $\epsilon^n = \omega(1/N)$ and $\epsilon = o(1/\sqrt{\log N})$, then we still have $T \to \infty$, but as $e^{-T^2/2} = \Theta\left(\frac{1}{\epsilon^n(N-1)}\right)$, we can see that the hit count increases with N:

$$E[\#\text{hits}] = \Omega\left(\epsilon^{-n}(\log(\epsilon^n N))^{\frac{n-2}{2}}\right)$$

If $\epsilon^n = \Theta(1/N)$, then $T = \Theta(1)$ and Equation 6.4 becomes invalid, because r does not grow to infinity in Equation 6.3 anymore: In this case, the probability term $P[||M|| \leq ...]$ in Equation 6.2 converges to a constant and we have linear growth:

$$E[\#\text{hits}] = \Omega\left(N\right)$$

And if $\epsilon^n = o(1/N)$, the probability term $P[||M|| \leq ...]$ in Equation 6.2 converges to one and as $N \to \infty$ we have[3]:

$$E[\#\text{hits}] \gtrsim \bar{Q}N$$

which, as it is valid for any $\bar{Q}$, ultimately means:

$$E[\#\text{hits}] \sim N.$$

6.2.2 Power law case

First of all, note that differently from the Gaussian case, in the power law case $\arg\max p_1(x_1) \neq 0$, because $p(x) = 0$ if $||x|| < m$. Rather, $p_1(x_1)$ increases as $||x_1||$ grows, achieving a maximum at $||x_1|| = m^*(\epsilon) \triangleq \|\arg\max_{x_1} p_1(x_1)\| \approx m$, and then decreases as $||x_1|| \to \infty$. Similarly to the Gaussian case, we will restrain

[2]Note that the previous chapter predicted the hit count of the "max-prob" method. Therefore, it works as a lower bound to the result of the "max-expect" algorithm when we are interested in the expected hit rate.

[3]We use the notation $f(N) \gtrsim g(N)$ to denote $f(N) \geq \tilde{g}(N)$ for some $\tilde{g}$ satisfying $\tilde{g}(N) \sim g(N)$, i.e.: $(\forall\gamma > 0)(\exists\bar{N}) : (N > \bar{N}) \Rightarrow f(N) \geq (1-\gamma)g(N)$.

$M > m^*(\epsilon) + \frac{3}{2}\epsilon\bar{r}$, so that $\max_{||y-M||<\frac{3}{2}\epsilon\bar{r}} p_1(y) = p_1(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r})$, obtaining:

$$P[Q(M) \geq \bar{Q}] \geq P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \left\{\max_{||y-M||<\frac{3}{2}\epsilon\bar{r}} p_1(y)\right\} \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right)\right]$$

$$\geq P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n p_1\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right) \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right) \ \wedge \ ||M|| > m^*(\epsilon) + \frac{3}{2}\epsilon\bar{r}\right].$$

In the power law model, we cannot compute $p_1(x)$ and $p_m(x)$ directly, so we will bound them using $p(x)$. Recall that for this model, $\lim_{||x_1||\to\infty} p_1(x_1)/p(x_1) = 1$ (as seen in Section 5.3.3), so we can write:

$$(\forall\gamma > 0)(\exists\bar{M} > m^*(\epsilon) + \frac{3}{2}\epsilon\bar{r}) :$$

$$\begin{aligned}(||M|| > \bar{M}) \Rightarrow p_1\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right) &< (1+\gamma)p\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right) \\ &= (1+\gamma)\frac{(n-\alpha)m^{\alpha-n}}{A_n}(||M|| - \frac{3}{2}\epsilon\bar{r})^{-\alpha}\end{aligned} \tag{6.5}$$

implying that there exist[4] constants $\bar{M}$ and β such that:

$$P\left[Q(M) \geq \bar{Q}\right] \geq$$

$$P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \beta\left(||M|| - \frac{3}{2}\epsilon\bar{r}\right)^{-\alpha} \leq \frac{1 - \frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right) \ \wedge \ ||M|| > \bar{M}\right]$$

$$= P\left[\left(||M|| - \frac{3}{2}\epsilon\bar{r}\right)^{-\alpha} \leq \frac{C/\epsilon^n}{(N-1)} \ \wedge \ ||M|| > \bar{M}\right]$$

Therefore:

$$E[\#\text{hits}] \geq N\bar{Q}P\left[||M|| \geq \frac{3}{2}\epsilon\bar{r} + \left(\frac{C/\epsilon^n}{N-1}\right)^{-\frac{1}{\alpha}} \ \wedge \ ||M|| > \bar{M}\right] \tag{6.6}$$

Now using that, for some $\tilde{\gamma}$,

$$P[||M|| \geq t] = \int_{||M||>t} p_m(M)dM \geq \int_{||M||>t}(1-\tilde{\gamma})p(M)dM = (1-\tilde{\gamma})(m/t)^{\alpha-n},$$

[4]This is particularly possible because $\epsilon = O(1)$ and $m^*(\epsilon) = O(1)$ as $N \to \infty$, ensuring β and $\bar{M}$ are constants with respect to ϵ and N.

we obtain, with $\tilde{C} = (1-\tilde{\gamma})m^{\alpha-n}$:

$$E[\#\text{hits}] \geq N\bar{Q}\tilde{C}\left(\frac{3}{2}\epsilon\bar{r} + \left(\frac{C/\epsilon^n}{N-1}\right)^{-1/\alpha}\right)^{n-\alpha}$$

$$= N\bar{Q}\tilde{C}\epsilon^{n-\alpha}\left(\frac{3}{2}\bar{r} + \epsilon^{n/\alpha-1}\left(\frac{C}{N-1}\right)^{-1/\alpha}\right)^{n-\alpha}$$

which, because $\epsilon^{n/\alpha-1} \to \infty$ and $\left(\frac{C}{N-1}\right)^{-1/\alpha} \to \infty$, is equal to:

$$\Theta(N\epsilon^{n-\alpha}(N^{1/\alpha}\epsilon^{n/\alpha-1})^{n-\alpha})$$

$$= \Theta(N(N\epsilon^n)^{(n-\alpha)/\alpha})$$

$$= \Theta(N^{n/\alpha}\epsilon^{n(n-\alpha)/\alpha})$$

Therefore, power law distributions have as lower bound $E[\#\text{hits}] = \Omega\left(N^{n/\alpha}\epsilon^{n(n-\alpha)/\alpha}\right)$. This is in accordance with the result from the previous chapter, that predicted an infinite hit count as $N \to \infty$ (i.e. $E[\#\text{hits}] = \omega(1)$).

Notice however that, similarly to the Gaussian model, if $\epsilon^n = \Theta(1/N)$ or $\epsilon^n = o(1/N)$, the term $\frac{3}{2}\epsilon\bar{r} + \left(\frac{C/\epsilon^n}{N-1}\right)^{-\frac{1}{\alpha}}$ in Equation 6.6 will respectively converge to a constant or decrease, and Equation 6.5 cannot be used. In these cases, we have respectively $E[\#\text{hits}] = \Omega(N)$ and $E[\#\text{hits}] \sim N$, as in the Gaussian case.

6.2.3 Exponential case

As in the other models, in the exponential model we will restrain $||M|| \geq \frac{3}{2}\epsilon\bar{r}$. Also, similarly to the power law case, we will use the fact that $\lim_{||x_1||\to\infty} p_1(x_1)/p(x_1) = e^{\lambda^2\epsilon^2/2}$ (as seen in Section 5.3.2) to write:

$$(\forall\gamma > 0)(\exists\bar{M} > \frac{3}{2}\epsilon\bar{r}) :$$

$$(||M|| > \bar{M}) \Rightarrow p_1\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right) < (1+\gamma)e^{\lambda^2\epsilon^2/2}p\left(M - \frac{3}{2}\frac{M}{||M||}\epsilon\bar{r}\right)$$
$$= (1+\gamma)e^{\lambda^2\epsilon^2/2}e^{\lambda\frac{3}{2}\epsilon\bar{r}}p(M)$$

so that there exist $\bar{M}$ and β such that:

$$P\left[Q(M) \geq \bar{Q}\right] \geq$$

$$P\left[\left(\frac{A_n}{n}\left(\frac{3}{2}\epsilon\bar{r}\right)^n \beta e^{\lambda^2\epsilon^2/2+\lambda\frac{3}{2}\epsilon\bar{r}}e^{-\lambda||M||} \leq \frac{1-\frac{\bar{Q}}{P[||D||<\epsilon\bar{r}]}}{2(N-1)}\right) \ \wedge \ ||M|| > \bar{M}\right]$$

$$= P\left[||M|| \geq \lambda\epsilon^2/2 + \frac{3}{2}\epsilon\bar{r} - \frac{1}{\lambda}\log\left(\frac{C/\epsilon^n}{(N-1)}\right) \ \wedge \ ||M|| > \bar{M}\right].$$

Using that

$$P[||M|| \geq t] = \int_{||M||>t} p_m(M)dM \geq \int_{||M||>t}(1-\tilde{\gamma})e^{\lambda^2\epsilon^2/4}p(M)dM$$

$$\geq \tilde{\beta}e^{\lambda^2\epsilon^2/4}e^{-\lambda t}(\lambda t)^{n-1}$$

with $t = \lambda\epsilon^2/2 + \frac{3}{2}\epsilon\bar{r} - \frac{1}{\lambda}\log\left(\frac{C/\epsilon^n}{(N-1)}\right)$ we obtain:

$$E[\#\text{hits}] \geq N\bar{Q}\tilde{\beta}e^{\lambda^2\epsilon^2/4}e^{-\lambda t}(\lambda t)^{n-1}$$

$$= N\frac{C/\epsilon^n}{(N-1)}\bar{Q}\tilde{\beta}e^{-\frac{3}{2}\lambda\epsilon\bar{r}-\lambda^2\epsilon^2/4}\left(\lambda^2\epsilon^2/2 + \frac{3}{2}\lambda\epsilon\bar{r} - \log\left(\frac{C/\epsilon^n}{(N-1)}\right)\right)^{n-1}$$

$$= \Theta\left(\frac{1}{\epsilon^n}(\log(N\epsilon^n))^{n-1}\right)$$

This means that we have found the lower bound $E[\#\text{hits}] = \Omega\left(\frac{1}{\epsilon^n}(\log(N\epsilon^n))^{n-1}\right)$ for the exponential distribution. This bound however is not tight: Experiments (Section 6.5.2) suggest that for fixed ϵ the asymptotic behavior is $\Theta((\log N)^n)$.

Also, note that the same phenomenon observed in the Gaussian and power law cases applies here when $\epsilon^n = \Theta(1/N)$ or $\epsilon^n = o(1/N)$, for the same reason of the power law case.

6.3 Condition for constant hit rate

As we have seen, $1/N$ appears to be a threshold function for ϵ^n with respect to expected hit rate, in such a way that if $\epsilon^n = \Theta(1/N)$ we have a minimum expected hit rate and if $\epsilon^n = o(1/N)$ we have $E[\#\text{hits}] \sim N$, i.e. a hit rate of 100% as $N \to \infty$. We can show that in fact this happens to almost any generator set distribution, not only Gaussian, power law or exponential distributions.

First of all, we will use that:

$$\max_{||y-M||<\frac{3}{2}\bar{r}\epsilon} p_1(y) \leq \max_{y\in\mathbb{R}^n} p_1(y) \leq \max_{y\in\mathbb{R}^n} p(y).$$

The second inequality above is valid because:

$$\forall y : p_1(y) = \{p * g_\epsilon\}(y) = \int_{\mathbb{R}^n} p(y-x)g_\epsilon(x)dx \leq \int_{\mathbb{R}^n}\left(\max_z p(z)\right)g_\epsilon(x)dx = \max_z p(z).$$

Therefore we can bound:

$$E[\#\text{hits}] \geq N\bar{Q}P\left[\frac{A_n}{n}\left(\frac{3}{2}\bar{r}\epsilon\right)^n \left\{\max_{y\in\mathbb{R}^n} p(y)\right\} \leq \frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{2(N-1)}\right]$$

$$= \begin{cases} \bar{Q}N & , \text{ if } \frac{A_n}{n}\left(\frac{3}{2}\bar{r}\epsilon\right)^n \{\max_{y\in\mathbb{R}^n} p(y)\} \leq \frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{2(N-1)}; \\ 0 & , \text{ otherwise.} \end{cases}$$

When $\epsilon^n = o(1/N)$, the condition becomes

$$\max_{y\in\mathbb{R}^n} p(y) \leq \frac{n}{A_n}\left(\frac{2}{3\epsilon\bar{r}}\right)^n \frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{2(N-1)} \sim +\infty \text{ (as } N\to\infty)$$

implying that, as long as $\max_{y\in\mathbb{R}^n} p(y) < +\infty$, we have $E[\#\text{hits}] \gtrsim \bar{Q}N$ for any $\bar{Q}$, and therefore $E[\#\text{hits}] \sim N$.

If $\epsilon^n \sim C/N$, the condition becomes:

$$\max_{y\in\mathbb{R}^n} p(y) \lesssim K\frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{C\bar{r}^n} \text{ (as } N\to\infty)$$

$$\text{where } K = \frac{1}{2}\frac{n}{A_n}\left(\frac{2}{3}\right)^n$$

which means we have a hit rate of at least $\bar{Q}$, as long as the condition is satisfied for that value of C.

We can show that the inequation above has the following properties:

- For every $C > 0$, there exists $\bar{Q} > 0$ such that the inequation is satisfied, which means that $[\epsilon^n = \Theta(1/N) \Rightarrow E[\#\text{hits}] = \Omega(N)]$;
- Similarly, for every $\bar{Q} \in]0,1[$, there exists a constant $C > 0$ that satisfies the inequation (thus guaranteeing a minimal hit rate of $\bar{Q}$).

For the first case, we can choose $\bar{r} = \left(\frac{K}{3C\{\max_y p(y)\}}\right)^{1/n}$, so that $\bar{Q} = \frac{1}{2}P[||D|| < \bar{r}\epsilon]$ satisfies the inequation:

$$\max_{y\in\mathbb{R}^n} p(y) \lesssim \frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{\left(\frac{1}{3\{\max_y p(y)\}}\right)} = \frac{3}{2}\{\max_y p(y)\}$$

For the second case, we can choose $\bar{r}$ such that $P[||D|| < \bar{r}\epsilon] = (\bar{Q}+1)/2$ and $C = \frac{1}{\bar{r}^n}K\left(\frac{1}{2}\{\max_y p(y)\}^{-1}\left(\frac{1-\bar{Q}}{1+\bar{Q}}\right)\right)$ satisfies the inequation:

$$\max_{y\in\mathbb{R}^n} p(y) \lesssim \frac{1-\frac{\bar{Q}}{P[||D||<\bar{r}\epsilon]}}{\frac{1}{2}\{\max_y p(y)\}^{-1}\left(\frac{1-\bar{Q}}{1+\bar{Q}}\right)} = 2\{\max_{y\in\mathbb{R}^n} p(y)\}$$

Notice that the only requirement on $p(x)$ for this condition for minimum hit rate ($\epsilon^n = O(1/N)$) to apply is that $\max_x p(x) < +\infty$. So for instance if $p(x)$ is a Dirac delta function, the threshold does not apply (and indeed we know that in this case, $E[\#\text{hits}] = 1$ regardless of N and ϵ).

Curiously, a related problem to that of probabilistic point matching, which we call *probabilistic point querying*, has the same property for minimum hit rate (see Appendix F).

6.4 Case with outliers

The previous derivations were done supposing there are no outliers ($q = 0$). The case with outliers is mostly analogous.

First of all, a term $(1 - q)$ is added multiplying the bound since we can only match x_1 and x_2 correctly if they are an inlier pair.

Secondly, the $B(M, D)$ function is different because the other pairs can also be outliers. However, the bound is the same: We write

$$B(M, D) = (1 - q)B_{\text{inlier}}(M, D) + qB_{\text{outlier}}(M, D)$$

where

$$B_{\text{inlier}}(M, D) = \int_{\mathbb{R}^n} P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D||\middle|\tilde{x}, D, M\right]^2 \text{pdf}[\tilde{x}]d\tilde{x}$$

$$B_{\text{outlier}}(M, D) = \left(\int_{\mathbb{R}^n} P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D||\middle|\tilde{x}, D, M\right] \text{pdf}[\tilde{x}]d\tilde{x}\right)^2$$

$$= P\left[||\tilde{x}_1 - M|| > \frac{3}{2}||D||\middle|D, M\right]^2$$

We have already shown that

$$1 - B_{\text{inlier}}(M, D) \leq 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D||\middle|D, M\right]$$

On the other side,

$$1 - B_{\text{outlier}}(M, D) = 1 - \left(1 - P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D||\middle|D, M\right]\right)^2$$

$$\leq 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D||\middle|D, M\right]$$

Therefore, the final bound on $B(\cdot, \cdot)$ is the same:

$$1 - B(M, D) \leq 2P\left[||\tilde{x}_1 - M|| < \frac{3}{2}||D||\middle|D, M\right]$$

The rest of the derivation remains unchanged, so that all the asymptotic behavior results are simply multiplied by $(1-q)$.

As for the condition for minimum hit rate, we have that for every $C > 0$ there exists $\bar{Q} \in]0,1[$, and for every $\bar{Q} \in]0,1[$ there exists $C > 0$ such that $\epsilon^n \sim C/N$ guarantees a minimum expected hit rate of $(1-q)\bar{Q}$ as $N \to \infty$.

6.5 Experiments

6.5.1 Power law asymptotic behavior

The purpose of this synthetic experiment is to validate our theoretic result on the asymptotic behavior of the hit count for power law distributions, i.e. confirm that $E[\#\text{hits}] = \Omega((N\epsilon^{n-\alpha})^{n/\alpha})$. Observing that $\lim_{N\to\infty} \frac{\log(\Omega(N^k))}{\log N} \geq k$, we will analyze the behavior of $\frac{\log(E[\#\text{hits}])}{\log N}$ as $N \to \infty$ and compare them to the theoretical value (lower bound). Matching is done using the Greedy #2 algorithm. We used 10 samples per case when $n = 1$ and 2 samples per case when $n > 1$.

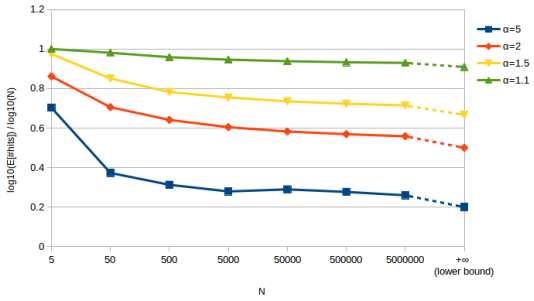
We first set $n = 1$, $m = 1$, $\epsilon = .25$, and vary $N \in \{5, 50, 500, ..., 5 \cdot 10^6\}$ and $\alpha \in \{1.1, 1.5, 2, 5\}$. The behavior of $\frac{\log(E[\#\text{hits}])}{\log N}$ converges to the theoretical value $(1/\alpha)$ as $N \to \infty$ (Figure 6.2(a)). If we use instead $\epsilon = 1/\sqrt{N}$, the theoretical value becomes $\frac{1}{2} + \frac{1}{2\alpha}$, which is also in agreement with the experiment (Figure 6.2(b)). The graphs suggest that the asymptotic bound is tight for $n = 1$, i.e., $E[\#\text{hits}] = \Theta((N\epsilon^{n-\alpha})^{n/\alpha})$ when $n = 1$.

For $n = 2$, we cannot analyze very high values of N, so we varied $N \in \{2, 4, 8, 16, ..., 2^9\}$. Fixing $m = 1$ and $\epsilon = .25$ and varying $\alpha \in \{2.2, 3, 4, 10\}$, the theoretic bound of $2/\alpha$ is also observed (Figure 6.2(c)), although it remains unclear whether the bound is tight or not.

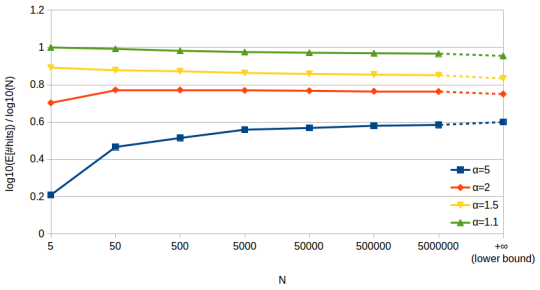
6.5.2 Exponential and Gaussian cases

The results from the previous chapter show that, for $\epsilon = \Theta(1)$, the asymptotic bound developed in this chapter for exponential and Gaussian distributions is loose: in the previous chapter we had observed that $E[\#\text{hits}] = \Theta(1)$ for Gaussian distributions and apparently $E[\#\text{hits}] = \Theta(\log N)$ for exponential distributions when $n = 1$, while the Gaussian lower bound converges to zero as $N \to \infty$ and the exponential one is $\Omega(1)$ when $n = 1$.

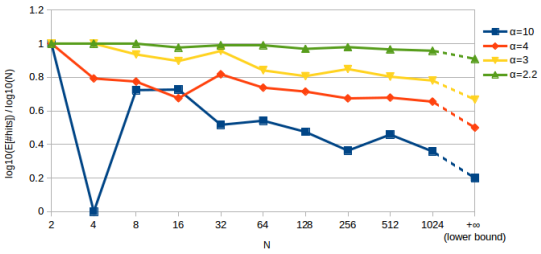
Running the Greedy #2 algorithm with $n = 2$, $\lambda = 1$ and fixed $\epsilon = .25$ suggests a squared-logarithmic behavior for the hit count with the exponential distribution (Figure 6.3(a)). This means that, while the lower bound of $\Omega((\log N)^{n-1})$ is correct, the actual behavior is most likely $\Theta((\log N)^n)$.



(a) $n = 1$, $\epsilon = \Theta(1)$

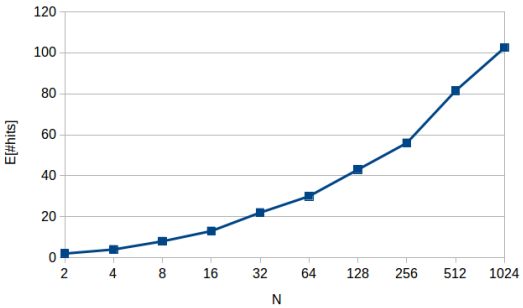


(b) $n = 1$, $\epsilon = \Theta(1/\sqrt{N})$

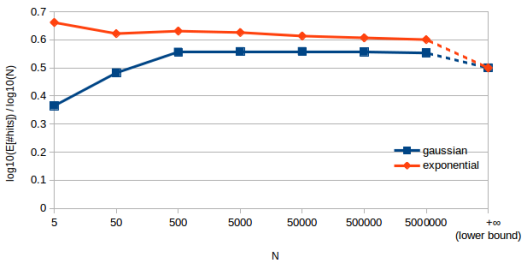


(c) $n = 2$, $\epsilon = \Theta(1)$

Figure 6.2: Behavior of $\frac{\log(E[\#\text{hits}])}{\log N}$ for power law distributions, also showing the theoretical bound in the end of the x-axis.



(a) Exponential distribution with $n = 2$ and $\epsilon = \Theta(1)$ shows a squared-logarithmic behavior for the hit count.



(b) Behavior of $\frac{\log(E[\#\text{hits}])}{\log N}$ for exponential and Gaussian distribution, with $n = 1$, $\epsilon = \Theta(1/\sqrt{N})$. Also showing the theoretical bound in the end of the x-axis.

Figure 6.3: Asymptotic behavior of exponential and Gaussian distributions

If we use $\epsilon = 1/\sqrt{N}$, with $n = 1$, the asymptotic bounds become $E[\#\text{hits}] = \Omega(\sqrt{N})$ for the exponential distribution and $E[\#\text{hits}] = \Omega(\sqrt{N}(\log N)^{-1/2})$ for the Gaussian distribution. Therefore, in both cases we should observe that $\lim_{N\to\infty} \frac{\log(E[\#\text{hits}])}{\log N} \geq 0.5$, which is consistent with the experiment of Figure 6.3(b) (using $\sigma = 1$ for the Gaussian distribution).

6.5.3 Constant hit rate

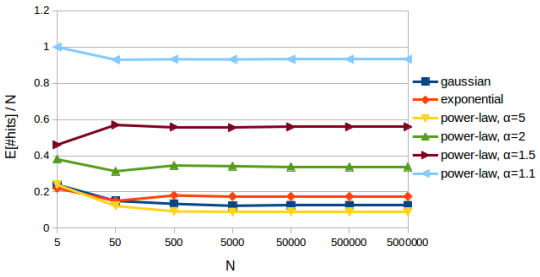
In this experiment we evaluate the condition for constant hit rate, $\epsilon^n = O(1/N)$.

We run Greedy #2 for multiple distributions with $\epsilon = 10/N$ for $n = 1$ and $\epsilon = 5/\sqrt{N}$ for $n = 2$. Figure 6.4(a,b) shows that the hit rate converges to a constant with different distributions (Gaussian, exponential, and power law for different values of α).

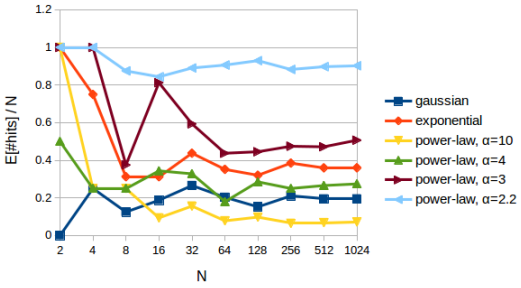
Next we see what happens if we use $\epsilon = C/N$ for different values of $C \in \{.1, 1, 10, 100\}$, for a Gaussian distribution with $n = 1$ and $\sigma = 1$. Our theoretic result is that, for every C there exists $\bar{Q}$, and for every $\bar{Q}$ there exists C such that $\epsilon^n \leq C/N \Rightarrow \frac{E[\#hits]}{N} \gtrsim \bar{Q}$, i.e. there is a bijective relation between C and $\bar{Q}$. Figure 6.4(c) evinces this relationship, as $C \in [.1, 100]$ already covers most of the values of $\bar{Q} \in (0, 1)$.

6.5.4 Greedy #2 and "max-prob"

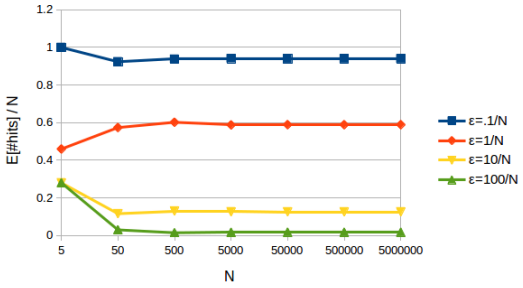
While the experiments of this section were done with Greedy #2, the results with "max-prob" are not much different. Figure 6.4(d) illustrates this, showing that the relationship between C and $\bar{Q}$, where $\epsilon^n = C/N$ and $n = 1$, is approximately the same for both algorithms, with slightly higher performance for "max-prob" around $C \approx 1$.



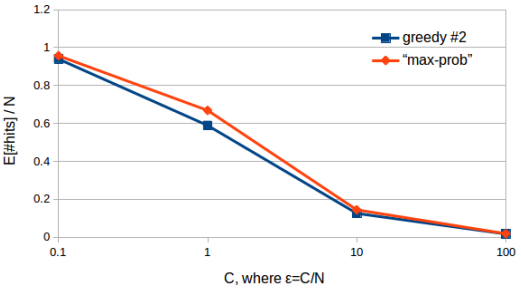
(a) $n = 1$, varied distributions



(b) $n = 2$, varied distributions



(c) $n = 1$, varying C in $\epsilon^n = C/N$ (Gaussian distribution)



(d) same as (c), but fixing $N = 5 \cdot 10^6$ and comparing Greedy #2 and "max-prob" algorithms

Figure 6.4: Hit rate with $\epsilon^n = \Theta(1/N)$.

Chapter 7

Matching All Pairs

Similarly to the previous chapter, that derived bounds for the asymptotic behavior of the expect hit count, this chapter analyzes the asymptotic behavior of the probability of matching all pairs correctly.

While the results from the previous chapter apply to Greedy #2 and consequently also to the "max-expect" method, but not necessarily to "max-prob"; the lower bounds from this chapter are also based on Greedy #2 and apply to "max-prob", as it is the the best method for this metric (i.e. it was designed to find the most probable permutation, and therefore maximizes the probability of matching all pairs correctly), but not necessarily to the "max-expect" method.

Similarly to the previous chapter, in this chapter we consider only the generator set model, and with isotropic Gaussian noise of parameter ϵ.

7.1 Condition for constant probability

In the same way that we showed that we can guarantee a minimum hit rate of $\bar{Q}$ if $\epsilon^n = C/N$ as $N \to \infty$, for some $C > 0$, we will show that a similar condition can guarantee a minimum probability of matching all points correctly as $N \to \infty$.

Again, we know that Greedy #2 matches a pair (x_1, x_2) correctly if all other points $\tilde{x}_1 \in P_1 \setminus \{x_1\}$ are farther from x_2 than x_1 and all $\tilde{x}_2 \in P_2 \setminus \{x_2\}$ are farther from x_1 than x_2. If this applies to all points in both sets, then the algorithm will have matched all pairs correctly. Therefore the probability that this condition is satisfied is less than or equal to the probability P_{all} of matching all pairs correctly with Greedy #2.

Suppose, without loss of generality, that $\Pi = I$. Let then M_i and D_i be the mean and difference vectors of the pair (X_1^i, X_2^i), for $i = 1, ..., N$. Let us bound the probability of all pairs being the closest to each other as greater or equal to the probability, for all pairs (i, j), that M_i and M_j have a distance of more than

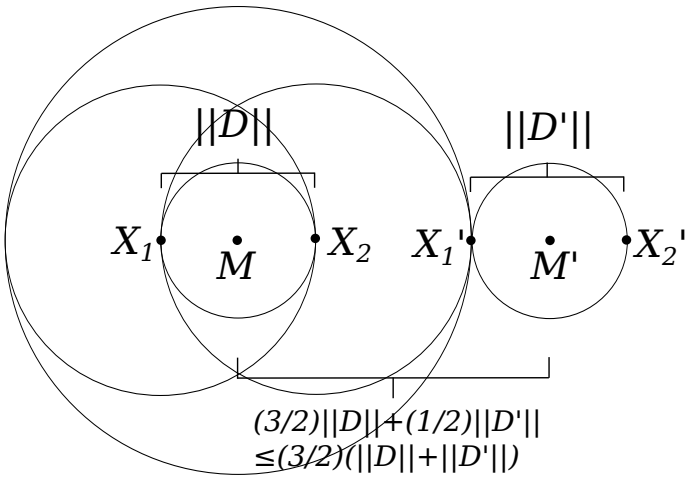


Figure 7.1: Illustration of the $\frac{3}{2}(||D_i|| + ||D_j||)$ safety radius.

$\frac{3}{2}(||D_i|| + ||D_j||)$, so that they do not conflict (see Figure 7.1), i.e.:

$$P_{\text{all}} \geq \tilde{P}_{\text{all}} \triangleq P\left[\bigwedge_{1\leq j<i\leq N} ||M_i - M_j|| > \frac{3}{2}(||D_i|| + ||D_j||)\right],$$

where "$\bigwedge$" denotes logical conjunction (the "and" operator).

Let $P[\text{out}_i]$ be the probability that the i-th pair does not conflict with the $i-1$ previous pairs, i.e.:

$$\text{out}_i \leftrightarrow \bigwedge_{j=1}^{i-1} ||M_i - M_j|| > \frac{3}{2}(||D_i|| + ||D_j||).$$

We have then:

$$\tilde{P}_{\text{all}} = P[\text{out}_2, \text{out}_3, .., \text{out}_N]$$

$$= \int P[\text{out}_2, \text{out}_3, .., \text{out}_N, D_1, ..., D_N] dD_1 ... dD_N$$

Applying recursively the rule $P[\text{out}_2, ..., \text{out}_k, D_1, ..., D_k] = P[\text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}].P[D_k, \text{out}_k | \text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}]$, we obtain:

$$\tilde{P}_{\text{all}} = \int P[D_1] P[\text{out}_2, D_2 | D_1] P[\text{out}_3, D_3 | D_1, D_2, \text{out}_2] ...$$

$$...P[\text{out}_4, D_4 | D_1, D_2, D_3, \text{out}_2, \text{out}_3] ... dD_1 ... dD_N$$

Now let us use that $P[\text{out}_k, D_k | \text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}] = P[\text{out}_k | \text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}, D_k].P[D_k | \text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}]$:

$$\tilde{P}_{\text{all}} = \int P[\text{out}_2 | D_1, D_2] P[\text{out}_3 | D_1, D_2, D_3, \text{out}_2] P[\text{out}_4 | D_1, ..., D_4, \text{out}_2, \text{out}_3] ...$$

$$...P[D_1] P[D_2 | D_1] P[D_3 | \text{out}_2, D_1, D_2] ... P[D_N | D_1, \text{out}_2, D_2, ..., \text{out}_{N-1}, D_{N-1}] dD_1 ... dD_N$$

Noting that $P[D_k|\text{out}_2, ..., \text{out}_{k-1}, D_1, ..., D_{k-1}] = P[D_k]$, since D_k is independent from $D_1, ..., D_{k-1}$ and $\text{out}_2, ..., \text{out}_{k-1}$ (recall that out_i depends only on $M_1, ..., M_i$ and $D_1, ..., D_i$), we obtain:

$$\tilde{P}_{\text{all}} = \int P[\text{out}_2|D_1, D_2]P[\text{out}_3|D_1, D_2, D_3, \text{out}_2]P[\text{out}_4|D_1, ..., D_4, \text{out}_2, \text{out}_3]...$$

$$...P[D_1]P[D_2]...P[D_N]dD_1...dD_N.$$

Now bounding using the maximum probability density $p_0 = \max_{y\in\mathbb{R}^n} p(y)$ times the volume of the containing sphere, we can write:

$$P[\text{out}_i|D_1, ..., D_i, \text{out}_2, ..., \text{out}_{i-1}] \geq \max\left\{1 - p_0\frac{A_n}{n}\sum_{j=1}^{i-1}\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n, 0\right\},$$

implying

$$P_{\text{all}} \geq \int\left(\prod_{i=1}^{N}\max\left\{0, 1 - p_0\frac{A_n}{n}\sum_{j=1}^{i-1}\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n\right\}\right)P[D_1]...P[D_N]dD_1...dD_N$$

$$= \text{E}\left[\prod_{i=1}^{N}\left(1 - \min\left\{1, p_0\frac{A_n}{n}\sum_{j=1}^{i-1}\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n\right\}\right)\right] \tag{7.1}$$

Using that $\prod_i(1-p_i) \geq 1 - \sum_i p_i$ for $0 < p_i < 1$, we can bound the result above to:

$$P_{\text{all}} \geq \text{E}\left[1 - \sum_{i=1}^{N}\min\left\{1, p_0\frac{A_n}{n}\sum_{j=1}^{i-1}\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n\right\}\right] \tag{7.2}$$

$$\geq \text{E}\left[1 - p_0\frac{A_n}{n}\sum_{1\leq j<i\leq N}\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n\right]$$

$$= 1 - p_0\frac{A_n}{n}\sum_{1\leq j<i\leq N}\text{E}\left[\left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n\right]$$

$$= 1 - p_0\frac{A_n}{n}\frac{N(N-1)}{2}\text{E}\left[\left(\frac{3}{2}(||D_1|| + ||D_2||)\right)^n\right]$$

We only need to compute now $\text{E}\left[\left(\frac{3}{2}(||D_1|| + ||D_2||)\right)^n\right]$. Note however that:

$$\text{E}\left[\left(\frac{3}{2}(||D_1|| + ||D_2||)\right)^n\right] = \text{E}\left[\left(\frac{3}{2}(||\sqrt{2}\epsilon G_1|| + ||\sqrt{2}\epsilon G_2||)\right)^n\right] =$$

$$\epsilon^n \cdot \text{E}\left[\left(\frac{3}{\sqrt{2}}(||G_1|| + ||G_2||)\right)^n\right] = \phi_n\epsilon^n$$

where G_1 and G_2 are independent isotropic Gaussian variables in $\mathbb{R}^n$ with unitary variance ($\text{Cov}[G_1] = \text{Cov}[G_2] = I_{n\times n}$), and ϕ_n is a constant that depends only on

the number of dimensions n.

We obtain then:

$$P_{\text{all}} \geq 1 - p_0 \frac{A_n}{n} \frac{N(N-1)}{2} \phi_n \epsilon^n \tag{7.3}$$

Computation of ϕ_n

To compute ϕ_n, we will use first that:

$$\begin{aligned} &\int_{\mathbb{R}^n} \frac{e^{-\frac{||x||^2}{2}}}{(2\pi)^{n/2}} dx = \int_0^\infty \frac{e^{-\frac{r^2}{2}}}{(2\pi)^{n/2}} A_n r^{n-1} dr = 1 \Rightarrow \\ &\int_0^\infty r^k e^{-\frac{r^2}{2}} dr = \frac{(2\pi)^{\frac{k+1}{2}}}{A_{k+1}} \end{aligned} \tag{7.4}$$

Then ϕ_n can be derived as follows:

$$\begin{aligned} \phi_n &= \mathrm{E}\left[\left(\frac{3}{\sqrt{2}}(||G_1|| + ||G_2||)\right)^n\right] \\ &= \int_0^\infty \int_0^\infty \left(\frac{3}{\sqrt{2}}(x+y)\right)^n \frac{e^{-\frac{x^2}{2}-\frac{y^2}{2}}}{(2\pi)^n} A_n^2 x^{n-1} y^{n-1} dx dy \\ &= \int_0^\infty \int_0^\infty \sum_{l=0}^n \binom{n}{l} (3/\sqrt{2})^n x^l y^{n-l} \frac{e^{-\frac{x^2}{2}-\frac{y^2}{2}}}{(2\pi)^n} A_n^2 x^{n-1} y^{n-1} dx dy \\ &= \int_0^\infty \int_0^\infty \sum_{l=0}^n \binom{n}{l} (3/\sqrt{2})^n x^{n+l-1} y^{2n-l-1} \frac{e^{-\frac{x^2}{2}-\frac{y^2}{2}}}{(2\pi)^n} A_n^2 dx dy \\ &= \sum_{l=0}^n \binom{n}{l} \frac{(3/\sqrt{2})^n}{(2\pi)^n} A_n^2 \left(\int_0^\infty x^{n+l-1} e^{-\frac{x^2}{2}} dx\right)\left(\int_0^\infty y^{2n-l-1} e^{-\frac{y^2}{2}} dy\right) \\ &= \sum_{l=0}^n \binom{n}{l} \frac{(3/\sqrt{2})^n}{(2\pi)^n} A_n^2 \frac{(2\pi)^{\frac{n+l}{2}} (2\pi)^{\frac{2n-l}{2}}}{A_{n+l} A_{2n-l}} \\ &= \sum_{l=0}^n \binom{n}{l} \frac{(3/\sqrt{2})^n A_n^2 (2\pi)^{n/2}}{A_{n+l} A_{2n-l}}. \end{aligned}$$

7.1.1 $\epsilon^n \sim C/N^2$ case

Interestingly, according to Equation 7.3, the threshold function for ϵ^n with respect to the probability of matching all pairs correctly appears to be $1/N^2$.

If $\epsilon^n \sim C/N^2$, we can bound:

$$P_{\text{all}} \gtrsim 1 - p_0 \frac{A_n}{n} \frac{\phi_n}{2} C \text{ (as } N \to \infty)$$

And if $\epsilon^n = o(1/N^2)$, we have $P_{\text{all}} \sim 1$.

This result contrasts with the result of the previous chapter, that when $\epsilon^n =$

$o(1/N)$, we already have 100% hit rate. However, this does not mean that we have 100% probability of matching all pairs correctly. Guaranteeing that $E[\#\text{hits}] \sim N$ does not mean that the *miss count*, i.e. the number of incorrectly matched pairs, is $E[\#\text{miss}] \to 0$, but rather that $E[\#\text{miss}] = o(N)$, which is something completely different from guaranteeing 100% probability of matching all pairs correctly[1]. See Section 7.2 for details.

A tighter bound

In fact, if $\epsilon^n \sim C/N^2$, we can show that P_{all} can be asymptotically bounded to:

$$P_{\text{all}} \gtrsim \exp\left(-p_0 \frac{A_n}{n} \frac{\phi_n}{2} C\right) \text{ (as } N \to \infty)$$

This is because the bound between Equations 7.1 and 7.2 is too loose. When $\epsilon^n \sim C/N^2$, we can replace it with a tighter bound. Suppose we want to bound a value L defined as

$$L = E\left[\prod_{i=1}^{N} 1 - \theta_i\right]$$

for random variables $\theta_1, ..., \theta_N$, where θ_i is of the form $\theta_i = \min\{1, \tilde{\theta}_i\}$. We can rewrite L as

$$\begin{aligned} L &= E\left[\prod_{i=1}^{N} \exp(\log(1-\theta_i))\right] \\ &= E\left[\prod_{i=1}^{N} \exp(-\theta_i - O(\theta_i^2))\right] \\ &= E\left[\exp\left(\sum_{i=1}^{N}(-\theta_i - O(\theta_i^2))\right)\right] \\ &= E\left[\exp\left(-\sum_{i=1}^{N}\theta_i - O\left(\sum_{i=1}^{N}\theta_i^2\right)\right)\right], \end{aligned}$$

which, because θ_i is of the form $\theta_i = \min\{1, \tilde{\theta}_i\}$, satisfies:

$$L \geq \tilde{L} \triangleq E\left[\exp\left(-\sum_{i=1}^{N}\tilde{\theta}_i - O\left(\sum_{i=1}^{N}\tilde{\theta}_i^2\right)\right)\right]. \tag{7.5}$$

Now note that as $N \to \infty$, if the variance of the exponent goes to zero, its distribution converges to a Dirac delta and we can move the expectation operator to the exponent, i.e., for a random variable X, $\text{Var}[X] = 0 \Rightarrow \text{pdf}[X = x] =$

[1]Also, having asymptotically 100% probability of matching all pairs correctly does not necessarily mean that the miss count converges to zero either. For instance, suppose we have $1 - o(1)$ probability of matching all pairs correctly, and $o(1)$ probability of missing $\lceil aN \rceil$ pairs for some $a \in]0, 1[$; then the miss count would be $\lceil aN \rceil o(1) = o(N)$.

$\delta(x - E[X]) \Rightarrow E[\exp(X)] = \exp(E[X])$, so we could write:

$$\lim_{N\to\infty} \tilde{L} = \lim_{N\to\infty} \exp\left(-E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right] - O\left(E\left[\sum_{i=1}^{N} \tilde{\theta}_i^2\right]\right)\right) \tag{7.6}$$

In our case,

$$\tilde{\theta}_i = p_0 \frac{A_n}{n} \sum_{j=1}^{i-1} \left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n$$

$$\Rightarrow \sum_{i=1}^{N} \tilde{\theta}_i = p_0 \frac{A_n}{n} \sum_{1\le j<i\le N} \left(\frac{3}{2}(||D_i|| + ||D_j||)\right)^n$$

$$= p_0 \frac{A_n}{n} \epsilon^n \sum_{1\le j<i\le N} \left(\frac{3}{\sqrt{2}}(||G_i|| + ||G_j||)\right)^n$$

$$\Rightarrow E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right] = p_0 \frac{A_n}{n} \epsilon^n \frac{N(N-1)}{2} \phi_n$$

where $G_1, ..., G_N$ are i.i.d. Gaussian variables of zero mean and unitary variance, and $\epsilon^n \sim C/N^2$.

The transition between Equations 7.5 and 7.6 requires showing that $\lim_{N\to\infty} \mathrm{Var}\left[\sum_{i=1}^{N} \tilde{\theta}_i\right] = 0$ and $\lim_{N\to\infty} \mathrm{Var}\left[\sum_{i=1}^{N} \tilde{\theta}_i^2\right] = 0$. For that end, we will show that[2]:

- $\lim_{N\to\infty} E\left[\left(\sum_{i=1}^{N} \tilde{\theta}_i\right)^2\right] = \lim_{N\to\infty} E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right]^2$, and
- $\lim_{N\to\infty} E\left[\left(\sum_{i=1}^{N} \tilde{\theta}_i^2\right)^2\right] = 0$, which also implies that $\lim_{N\to\infty} E\left[\sum_{i=1}^{N} \tilde{\theta}_i^2\right] = 0$.

Let $F(x, y) = \left(\frac{3}{\sqrt{2}}(||x|| + ||y||)\right)^n$. We have then

$$E\left[\left(\sum_{i=1}^{N} \tilde{\theta}_i\right)^2\right] = \left(p_0 \frac{A_n}{n} \epsilon^n\right)^2 E\left[\left(\sum_{1\le j<i\le N} F(G_i, G_j)\right)^2\right]$$

$$\sim \left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 E\left[\left(\sum_{1\le j<i\le N} F(G_i, G_j)\right)^2\right]$$

$$= \left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 \frac{1}{4} E\left[\left(\sum_{i\neq j} F(G_i, G_j)\right)^2\right]$$

$$= \left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 \frac{1}{4} \sum_{i\neq j} \sum_{i'\neq j'} E\left[F(G_i, G_j) F(G_{i'}, G_{j'})\right]$$

[2]Showing that $\mathrm{Var}[X] = 0$ for some random variable X is the same as showing that $E[X^2] = E[X]^2$, since $E[X^2] = E[X]^2 + \mathrm{Var}[X]$.

Note that the expectation term in the expression above yields different values depending if $i \neq j \neq i' \neq j'$ or if there are indices in common. There are $N(N-1)(N-2)(N-3)$ terms with all four distinct indices, $\Theta(N^3)$ terms with three distinct indices and $\Theta(N^2)$ terms with two distinct indices. Therefore, the expression above is equal to:

$$\left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 \frac{1}{4} \left(\sum_{i \neq j \neq i' \neq j'} E\left[F(G_i, G_j) F(G_{i'}, G_{j'})\right] ... \right.$$

$$\left. ... + \sum_{\substack{i \neq j, i' \neq j', \\ |\{i\} \cup \{j\} \cup \{i'\} \cup \{j'\}| = 3}} E\left[F(G_i, G_j) F(G_{i'}, G_{j'})\right] + 2 \sum_{i \neq j} E\left[F(G_i, G_j) F(G_i, G_j)\right] \right)$$

$$\sim \left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 \frac{1}{4} \left(N^4 E[F(G_1, G_2) F(G_3, G_4)] + ... \right.$$

$$\left. ... \Theta(N^3) E[F(G_1, G_2) F(G_1, G_3)] + \Theta(N^2) E[F(G_1, G_2) F(G_1, G_2)] \right)$$

$$\sim \left(p_0 \frac{A_n}{n} \frac{C}{N^2}\right)^2 \frac{N^4}{4} E[F(G_1, G_2) F(G_3, G_4)]$$

$$= \left(p_0 \frac{A_n}{n} C\right)^2 \frac{1}{4} E[F(G_1, G_2)]^2$$

$$= \left(p_0 \frac{A_n}{n} C \frac{\phi_n}{2}\right)^2 = \lim_{N \to \infty} E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right]^2$$

as we wanted to demonstrate.

Meanwhile:

$$E\left[\left(\sum_{i=1}^{N} \tilde{\theta}_i^2\right)^2\right] = \left(p_0 \frac{A_n}{n}\right)^4 E\left[\left(\sum_{i=1}^{N} \left(\sum_{j=1}^{i-1} \epsilon^n F(G_i, G_j)\right)^2\right)^2\right]$$

$$= O(\epsilon^{4n} N^6) = O\left(\frac{1}{N^8} N^6\right) = O(1/N^2) \to 0.$$

Finally, we obtain:

$$\lim_{N \to \infty} \tilde{L} = \lim_{N \to \infty} \exp\left(-E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right] - O\left(E\left[\sum_{i=1}^{N} \tilde{\theta}_i^2\right]\right)\right)$$

$$= \lim_{N \to \infty} \exp\left(-E\left[\sum_{i=1}^{N} \tilde{\theta}_i\right]\right)$$

$$= \exp\left(-p_0 \frac{A_n}{n} \frac{\phi_n}{2} C\right)$$

$$\Rightarrow P_{\text{all}} \gtrsim \exp\left(-p_0 \frac{A_n}{n} \frac{\phi_n}{2} C\right).$$

This bound means that there is a bijective relation between P_{all} and C: For every $C > 0$, there is some constant $\bar{P}_{\text{all}} > 0$, and for every $\bar{P}_{\text{all}} \in]0, 1[$ there is $C > 0$ such that $\epsilon^n \sim C/N^2 \Rightarrow P_{\text{all}} \gtrsim \bar{P}_{\text{all}}$, similarly to the case seen in the last chapter (Section 6.5.3).

7.2 Experiments

7.2.1 Probability of matching all pairs correctly and miss count

Here we evaluate how the probability of matching all pairs correctly[3] and the miss count change with different behaviors of ϵ as $N \to \infty$, using the "max-prob" algorithm, with isotropic Gaussian distributions with $\sigma = 1$ and 100 samples per case.

First we do $\epsilon = C/N^{1.5}$, for $C \in \{.1, 1, 10, 100\}$, and $n = 1$. Because $\epsilon^n = o(1/N)$, the results from the previous chapter imply that $E[\#\text{hits}] \sim N$ and therefore the miss count is $E[\#\text{misses}] = N - E[\#\text{hits}] = o(N)$. Figure 7.2(a,b) shows that the probability of matching all pairs correctly goes to 0 as $N \to \infty$ (as expected), while the miss count is a power law of N, apparently $\Theta(\sqrt{N})$, which is $o(N)$.

If we use instead $\epsilon = C/N^2$, for the same range of values of C, we observe that the probability of hitting all points, as well as the miss count, converge to a constant as $N \to \infty$, for all values of C (Figure 7.2(c,d)). If we use instead $\epsilon = C/N^{2.5}$, the probability of hitting all points goes to 1 and the miss count converges to 0 Figure 7.2(e,f).

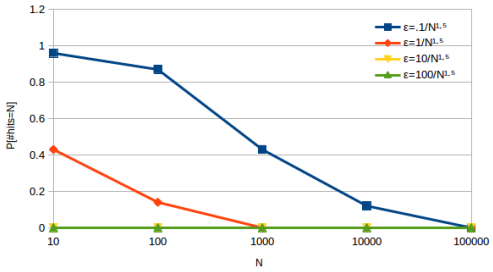
"Max-prob" and Greedy # 2

If we use Greedy #2 instead of "max-prob", results are similar, but "max-prob" has about 10% higher probability of hitting all points than Greedy #2. Figure 7.3(a,b) shows the case where $\epsilon = 1/N^2$, using 1000 samples per case.

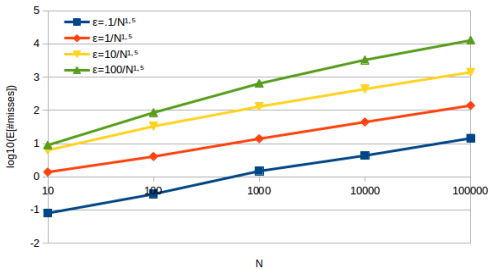
$n > 1$ case

The results are also confirmed in higher dimensions: With $n = 2$, using $\epsilon = 2/N$ and 100 samples per case, we see that the probability of hitting all points and the expected miss count converge to a constant, as expected (Figure 7.4(a,b)).

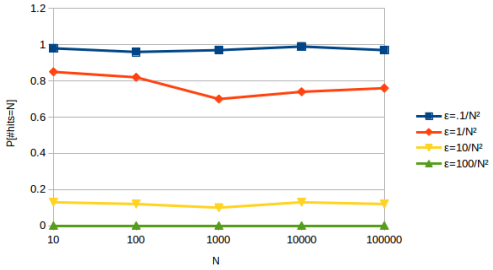
[3] estimated, i.e., we count the number of times that the algorithm matched all pairs correctly



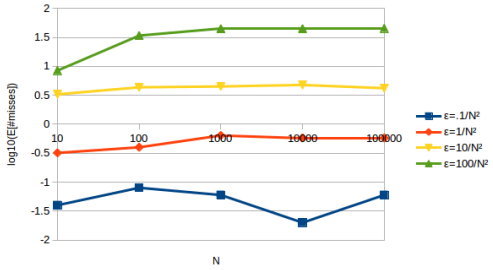
(a) Probability of hitting all points, $\epsilon = \Theta(1/N^{1.5})$



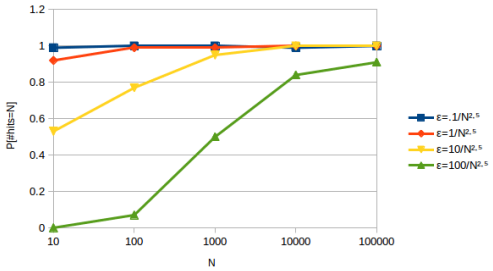
(b) Miss count, $\epsilon = \Theta(1/N^{1.5})$ (in logarithmic scale)



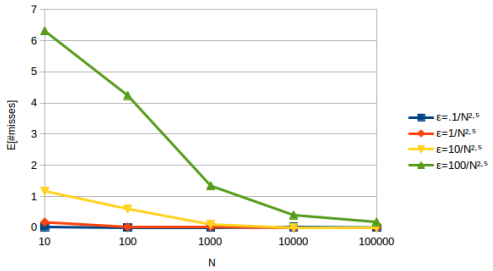
(c) Probability of hitting all points, $\epsilon = \Theta(1/N^2)$



(d) Miss count, $\epsilon = \Theta(1/N^2)$ (in logarithmic scale)

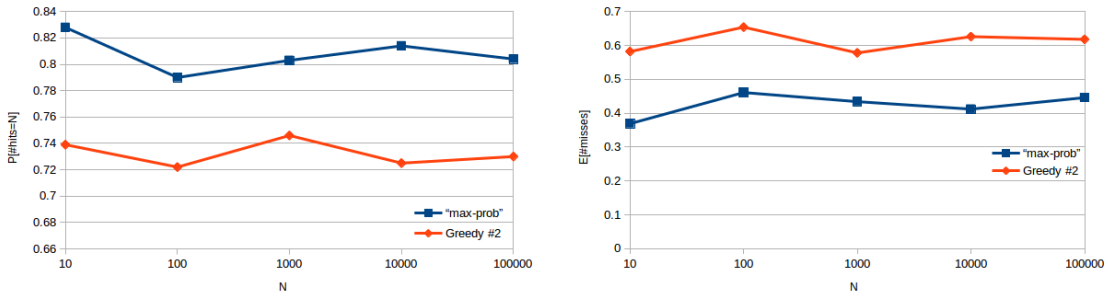


(e) Probability of hitting all points, $\epsilon = \Theta(1/N^{2.5})$



(f) Miss count, $\epsilon = \Theta(1/N^{2.5})$ (in normal scale)

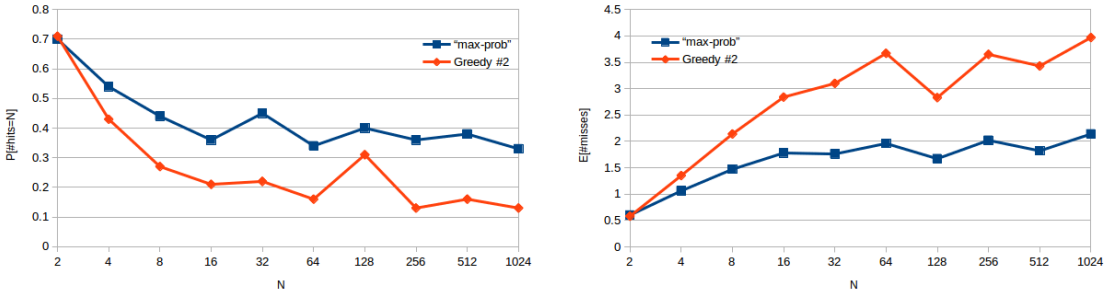
Figure 7.2: Asymptotic behavior of the probability of hitting all points and the miss count as $N \to \infty$, for $n = 1$, Gaussian distributions and different behaviors of ϵ as $N \to \infty$.



(a) Probability of hitting all points

(b) Miss count

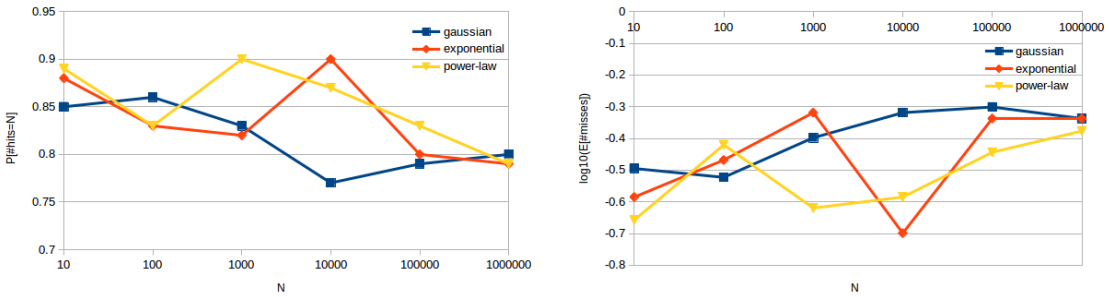
Figure 7.3: "max-prob" and Greedy #2 compared, $n = 1$, Gaussian distributions



(a) Probability of hitting all points

(b) Miss count

Figure 7.4: Results in $\mathbb{R}^2$, Gaussian distributions



(a) Probability of hitting all points

(b) Miss count

Figure 7.5: Comparing different distributions

7.2.2 Other distributions

With exponential and power law distributions we also observe that the probability of hitting all points converges to a constant. Figure 7.5(a,b) shows this behavior for $\epsilon = 1/N^2$, $n = 1$, using 100 samples per case, where the parameters of the Gaussian, exponential and power law distributions are respectively $\sigma = 1$, $\lambda = 1$, and $\alpha = 2$ and $m = 1$.

Part III

Application

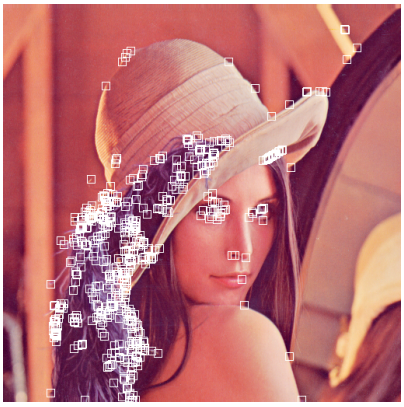
Chapter 8

Probabilistic models for image features

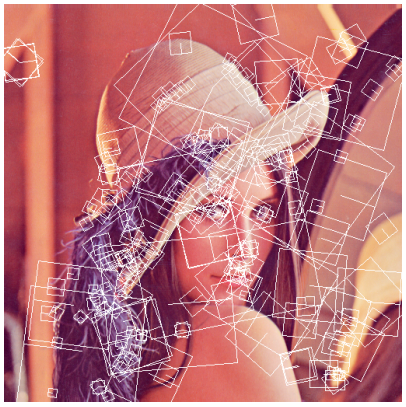
In this chapter, we introduce two probabilistic models for *feature matching* applications for use with our "max-prob" method: One for the Harris/NCC feature and the other for the SIFT feature.

The Harris/NCC case refers to the coupling of the Harris filter [27] (a feature detector) with Normalized Cross-Correlation [1] (a feature descriptor and comparison method), commonly used in applications such as 3D reconstruction and image stitching. The Harris filter detects corner-like points in images (Figure 8.1(a)) and NCC describes these points by taking an $L \times L$ pixel patch (represented as a vector $\tilde{x} \in \mathbb{R}^n$ with $n = 3L^2$, when there are 3 color channels) around the feature point, then normalizing it to satisfy zero mean and unitary norm (i.e., performing $x = \frac{\tilde{x} - \frac{\vec{1}\vec{1}^T}{n}\tilde{x}}{||\tilde{x} - \frac{\vec{1}\vec{1}^T}{n}\tilde{x}||}$, where $\vec{1} = \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$, so that x satisfies $\vec{1}^T x = 0$ and $||x|| = 1$), which provides the feature descriptor some robustness to illumination changes. The similarity between pairs of feature descriptors is given by the inner product between them ($\langle x, y \rangle$, which, because they satisfy $||x|| = ||y|| = 1$, is inversely related to the Euclidean distance between them, i.e. $||x-y||^2 = ||x||^2 - 2\langle x, y \rangle + ||y||^2 = 2 - 2\langle x, y \rangle$).

On the other hand, SIFT [15] is a more sophisticated feature detector and descriptor, commonly used in (but not limited to) recognition applications. SIFT has the advantage of being scale and rotation invariant: its blob-like feature points (Figure 8.1(b)) are all assigned a dominant direction (rotation) and the scale in which they were found. The descriptor is a $4 \times 4 \times 8$ (3D) histogram (forming a vector in $\mathbb{R}_+^{128}$) that stores the image gradients around the feature point according to their location and orientation relative to the feature point (in location, scale and orientation). Pairs of SIFT descriptors may be then compared using for instance Euclidean distance or Hellinger distance [28].



(a) Harris/NCC feature points and descriptors



(b) SIFT feature points and descriptors

Figure 8.1: Illustration of feature models.

8.1 Harris/NCC model

A probabilistic model for the Harris/NCC feature must satisfy that a feature from sets P_1 or P_2, which we denote as x_1, must satisfy:

$$||x_1||^2 = 1$$

$$x_1^T \vec{1} = 0$$

where $\vec{1} = \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}$. This is because the Harris/NCC descriptor subtracts pixel values from the mean value and scales values so that the squared sum equals 1. Note that these two restrictions imply that x_1 has $n-2$ degrees of freedom, and its PDF is defined on an $(n-2)$-dimensional subspace of $\mathbb{R}^n$.

8.1.1 Probabilistic model

In this model, we have three parameters: σ, ϵ and q, but in practice only two parameters need to be known: the ratio ϵ/σ and the outlier rate q.

The generating model is based on the generator set model with (asymmetric) outliers described in Section 3.3:

$$\tilde{X}_1 = X + Y_1$$

$$\tilde{X}_2 = ((XS + X'(I - S)) + Y_2)\Pi$$

with isotropic Gaussian distributions for the points in the generator set and for noise, with parameters σ and ϵ, and an outlier rate of q. The only difference is that we also project $\tilde{X}_1, \tilde{X}_2$ to X_1, X_2 to enforce the zero mean and unitary norm

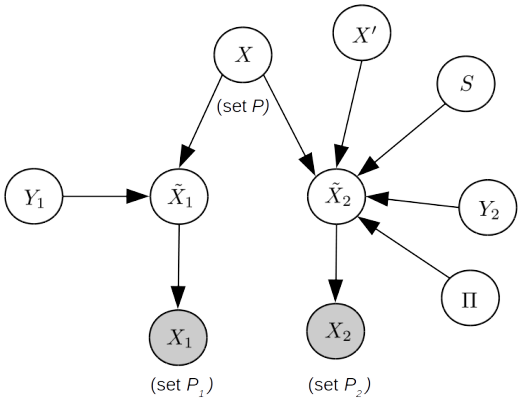


Figure 8.2: Bayesian network of the Harris/NCC feature probabilistic model.

constraints, following

$$X_1^i = \frac{(I - \frac{\vec{1}\vec{1}^T}{n})\tilde{X}_1^i}{\left\|(I - \frac{\vec{1}\vec{1}^T}{n})\tilde{X}_1^i\right\|} \tag{8.1}$$

$$X_2^i = \frac{(I - \frac{\vec{1}\vec{1}^T}{n})\tilde{X}_2^i}{\left\|(I - \frac{\vec{1}\vec{1}^T}{n})\tilde{X}_2^i\right\|}.$$

See Figure 8.2 for an illustration (Bayesian network) of the generative model.

8.1.2 Measure choice

The "max-prob" method fills the cost matrix using the function:

$$C(X_1^i, X_2^j) = -\log\left(\text{pdf}[X_1^i, X_2^j | \Pi_{ij} = 1]\right)$$

However, because a point $x_1 \in P_1$ is located in a degenerate subset (with $n-2$ degrees of freedom) of $\mathbb{R}^n$, the choice of the measure on x_1 affects the cost function.

Nevertheless, although the change of the measure results in a change of the cost function, the resulting matching will not be affected, because minimum bipartite matching is invariant to changes of the form $C(x,y) + f(x) + g(y)$ to the cost function. Suppose we defined $\text{pdf}[x_1] = \frac{dP}{d\mu(x_1)}$, for some measure μ. If we change it to μ', we obtain:

$$-\log\left(\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^i)}\right) =$$

$$-\log\left(\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu'(X_1^i)d\mu'(X_2^i)}\right) - \log\left(\frac{d\mu'(X_1^i)}{d\mu(X_1^i)}\right) - \log\left(\frac{d\mu'(X_2^i)}{d\mu(X_2^i)}\right)$$

which is of the form $C(x,y) + f(x) + g(y)$.

Our choice of measure for x_1 is the canonical measure for Riemannian manifolds (volume element): Suppose that the subset of $\mathbb{R}^n$ where x_1 is contained can be

parameterized by $n-2$ variables forming a vector $\theta \in \mathbb{R}^{n-2}$, i.e., $x_1 = x_1(\theta)$. Also, let $\partial_\theta x_1$ be the Jacobian of this parameterization. This measure $\mu(x_1)$ is defined as the one such that $\frac{d\mu(x_1)}{d\theta} = \sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}$, where $d\theta$ is the standard Lebesgue measure (Euclidean hyper-volume in $\mathbb{R}^{n-2}$).

This measure is invariant to the choice of θ: If we change the parameterization scheme to one that uses some other $n-2$ parameters $\theta' \in \mathbb{R}^{n-2}$, where $\theta' = \theta'(\theta)$, we obtain a measure $\mu'(x_1)$ where

$$\begin{aligned} d\mu'(x_1) &= \sqrt{\det((\partial_{\theta'} x_1)^T(\partial_{\theta'} x_1))}d\theta' \\ &= \sqrt{\det((\partial_{\theta'} x_1)^T(\partial_{\theta'} x_1))}\det(\partial_\theta \theta')d\theta \\ &= \sqrt{\det((\partial_\theta \theta')^T(\partial_{\theta'} x_1)^T(\partial_{\theta'} x_1)(\partial_\theta \theta'))}d\theta \\ &= \sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}d\theta \\ &= d\mu(x_1) \end{aligned}$$

Now let us see how this measure relates to $\tilde{x}_1$. We can write $\tilde{x}_1$ as:

$$\tilde{x}_1 = m_1 \frac{\vec{1}}{\sqrt{n}} + s_1 x_1$$

where $m_1/\sqrt{n}$ is the mean value of the components of $\tilde{x}_1$ and s_1 is the L2 norm of $\tilde{x}_1$ subtracted from its mean:

$$\frac{\vec{1}^T \tilde{x}_1}{n} = \frac{\vec{1}}{n}^T \left(m_1 \frac{\vec{1}}{\sqrt{n}} + s_1 x_1 \right) = \frac{m_1}{n} \frac{\vec{1}^T \vec{1}}{\sqrt{n}} = m_1/\sqrt{n}$$

$$\left\| \left(I - \frac{\vec{1}\vec{1}^T}{n} \right) \tilde{x}_1 \right\| = \left\| \left(I - \frac{\vec{1}\vec{1}^T}{n} \right) \left(m_1 \frac{\vec{1}}{\sqrt{n}} + s_1 x_1 \right) \right\| = ||s_1 x_1|| = s_1$$

Note also therefore that, by replacing $\tilde{x}_1$ with this expression in Equation 8.1, we obtain back x_1.

We can then write

$$\begin{aligned}
\frac{d\tilde{x}_1}{ds_1 dm_1 d\mu(x_1)} &= \frac{d\tilde{x}_1}{dm_1 ds_1 d\theta}\frac{d\theta}{d\mu(x_1)} \\
&= \left|\det\begin{bmatrix}\partial_{m_1}\tilde{x}_1 & \partial_{s_1}\tilde{x}_1 & \partial_\theta\tilde{x}_1\end{bmatrix}\right| \frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \\
&= \left|\det\begin{bmatrix}\partial_{m_1}\tilde{x}_1 & \partial_{s_1}\tilde{x}_1 & \partial_{x_1}\tilde{x}_1\partial_\theta x_1\end{bmatrix}\right| \frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \\
&= \left|\det\begin{bmatrix}\frac{\vec{1}}{\sqrt{n}} & x_1 & s_1.\partial_\theta x_1\end{bmatrix}\right| \frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \\
&= s_1^{n-2}\left|\det\begin{bmatrix}\frac{\vec{1}}{\sqrt{n}} & x_1 & \partial_\theta x_1\end{bmatrix}\right| \frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \\
&= s_1^{n-2}\sqrt{\det\left(\begin{bmatrix}\frac{\vec{1}^T}{\sqrt{n}} \\ x_1^T \\ \partial_\theta x_1^T\end{bmatrix}\begin{bmatrix}\frac{\vec{1}}{\sqrt{n}} & x_1 & \partial_\theta x_1\end{bmatrix}\right)}\frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}}
\end{aligned}$$

Note however that $\vec{1}^T x_1 = 0$ and $||x_1|| = 1$ implies that $\vec{1}^T\partial_\theta x_1 = 0$ and $x_1^T\partial_\theta x_1 = 0$, reducing the expression above to:

$$\begin{aligned}
\frac{d\tilde{x}_1}{ds_1 dm_1 d\mu(x_1)} &= s_1^{n-2}\sqrt{\det\left(\begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \partial_\theta x_1^T\partial_\theta x_1\end{bmatrix}\right)}\frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \Leftrightarrow \\
\frac{d\tilde{x}_1}{ds_1 dm_1 d\mu(x_1)} &= s_1^{n-2}\sqrt{\det\left(\partial_\theta x_1^T\partial_\theta x_1\right)}\frac{1}{\sqrt{\det((\partial_\theta x_1)^T(\partial_\theta x_1))}} \Leftrightarrow \\
\frac{d\tilde{x}_1}{ds_1 dm_1 d\mu(x_1)} &= s_1^{n-2}. \qquad (8.2)
\end{aligned}$$

8.1.3 Cost function

Now we can compute the cost function. Denoting $\text{pdf}[\tilde{X}_1^i, \tilde{X}_2^j|\Pi_{ij} = 1] = F(\tilde{X}_1^i, \tilde{X}_2^j)$, we derive from Equation 8.2 that[1]:

$$\begin{aligned}
C(x_1, x_2) &= -\log\left(\frac{dP[x_1, x_2|\Pi_{ij} = 1]}{d\mu(x_1)d\mu(x_2)}\right) \\
&= -\log\iiiint_{s_1,s_2>0}\text{pdf}[\tilde{x}_1, \tilde{x}_2|\Pi_{ij} = 1]\frac{d\tilde{x}_1}{ds_1 dm_1 d\mu(x_1)}\frac{d\tilde{x}_2}{ds_2 dm_2 d\mu(x_2)}ds_1 ds_2 dm_1 dm_2 \\
&= -\log\iiiint_{s_1,s_2>0} F\left(m_1\frac{\vec{1}}{\sqrt{n}} + s_1 x_1, m_2\frac{\vec{1}}{\sqrt{n}} + s_2 x_2\right)s_1^{n-2}s_2^{n-2}dm_1 dm_2 ds_1 ds_2
\end{aligned}$$

[1]By abuse of notation, $x_1 = X_1^i$, $x_2 = X_2^j$, $\tilde{x}_1 = \tilde{X}_1^i$ and $\tilde{x}_2 = \tilde{X}_2^j$.

Now let us take a closer look at $F(\tilde{x}_1, \tilde{x}_2)$. We know that

$$F(\tilde{x}_1, \tilde{x}_2) = g_{\sqrt{\sigma^2+\epsilon^2/2}}\left(\frac{\tilde{x}_1+\tilde{x}_2}{2}\right) g_{\sqrt{2}\epsilon}(\tilde{x}_1-\tilde{x}_2)(1-q) + g_{\sqrt{\sigma^2+\epsilon^2}}(\tilde{x}_1) g_{\sqrt{\sigma^2+\epsilon^2}}(\tilde{x}_2) q$$

$$\triangleq A(1-q) + Bq$$

Let us analyze first the "B" term.

$$B = \frac{\exp\left(-\frac{1}{2}\frac{\left\|m_1\frac{\vec{1}}{\sqrt{n}}+s_1x_1\right\|^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{\left\|m_2\frac{\vec{1}}{\sqrt{n}}+s_2x_2\right\|^2}{\sigma^2+\epsilon^2}\right)}{(2\pi(\sigma^2+\epsilon^2))^n}$$

As

$$\left\|m_1\frac{\vec{1}}{\sqrt{n}}+s_1x_1\right\|^2 = \left\|m_1\frac{\vec{1}}{\sqrt{n}}\right\|^2 + \|s_1x_1\|^2 = m_1^2+s_1^2$$

the expression for B simplifies to:

$$B = \frac{\exp\left(-\frac{1}{2}\frac{m_1^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{s_1^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{m_2^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{s_2^2}{\sigma^2+\epsilon^2}\right)}{(2\pi(\sigma^2+\epsilon^2))^n}$$

Note also that:

$$\iint_{\mathbb{R}^n}\frac{\exp\left(-\frac{1}{2}\frac{m_1^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{m_2^2}{\sigma^2+\epsilon^2}\right)}{(2\pi)(\sigma^2+\epsilon^2)^{1/2}(\sigma^2+\epsilon^2)^{1/2}}dm_1dm_2 = 1$$

and

$$\iint_{s_1,s_2>0}\frac{\exp\left(-\frac{1}{2}\frac{s_1^2}{\sigma^2+\epsilon^2}\right)\exp\left(-\frac{1}{2}\frac{s_2^2}{\sigma^2+\epsilon^2}\right)A_{n-1}^2 s_1^{n-2}s_2^{n-2}}{(2\pi)^{n-1}(\sigma^2+\epsilon^2)^{(n-1)/2}(\sigma^2+\epsilon^2)^{(n-1)/2}}ds_1ds_2 = 1$$

So the Bq term of the quadruple integral integrates to q/A_{n-1}^2.

Now the "A" term:

$$A = \frac{\exp(-\frac{1}{2}\frac{||(\tilde{x}_1+\tilde{x}_2)/2||^2}{\sigma^2+\epsilon^2/2})\exp(-\frac{1}{2}\frac{||\tilde{x}_1-\tilde{x}_2||^2}{2\epsilon^2})}{(2\pi)^n(\sigma^2+\epsilon^2/2)^{n/2}(2\epsilon^2)^{n/2}}$$

Note that

$$\left\|\frac{\tilde{x}_1+\tilde{x}_2}{2}\right\|^2 = \left\|\frac{m_1\frac{\vec{1}}{\sqrt{n}}+s_1x_1+m_2\frac{\vec{1}}{\sqrt{n}}+s_2x_2}{2}\right\|^2 = \left(\frac{m_1+m_2}{2}\right)^2 + \left\|\frac{s_1x_1+s_2x_2}{2}\right\|^2$$

and

$$\|\tilde{x}_1-\tilde{x}_2\|^2 = \left\|m_1\frac{\vec{1}}{\sqrt{n}}+s_1x_1-m_2\frac{\vec{1}}{\sqrt{n}}-s_2x_2\right\|^2 = (m_1-m_2)^2+\|s_1x_1-s_2x_2\|^2$$

so the expression for A reduces to:

$$A = \frac{\exp(-\frac{1}{2}\frac{((m_1+m_2)/2)^2}{\sigma^2+\epsilon^2/2})\exp(-\frac{1}{2}\frac{(m_1-m_2)^2}{2\epsilon^2})\exp(-\frac{1}{2}\frac{||(s_1x_1+s_2x_2)/2||^2}{\sigma^2+\epsilon^2/2})\exp(-\frac{1}{2}\frac{||s_1x_1-s_2x_2||^2}{2\epsilon^2})}{(2\pi)^n(\sigma^2+\epsilon^2/2)^{n/2}(2\epsilon^2)^{n/2}}$$

Because

$$\iint_{\mathbb{R}^n} \frac{\exp(-\frac{1}{2}\frac{((m_1+m_2)/2)^2}{\sigma^2+\epsilon^2/2})\exp(-\frac{1}{2}\frac{(m_1-m_2)^2}{2\epsilon^2})}{(2\pi)(\sigma^2+\epsilon^2/2)^{1/2}(2\epsilon^2)^{1/2}} dm_1 dm_2 = 1$$

the $A(1-q)$ term of the quadruple integral reduces to a double integral, yielding:

$$\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^j)} = \frac{q}{A_{n-1}^2} + (1-q)\iint_{s_1,s_2>0} ...$$

$$...\frac{\exp(-\frac{1}{2}\frac{||(s_1x_1+s_2x_2)/2||^2}{\sigma^2+\epsilon^2/2})\exp(-\frac{1}{2}\frac{||s_1x_1-s_2x_2||^2}{2\epsilon^2})}{(2\pi)^{n-1}(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}} s_1^{n-2}s_2^{n-2}ds_1ds_2$$

$$= \frac{q}{A_{n-1}^2} + (1-q)\iint_{s_1,s_2>0} ...$$

$$...\frac{\exp\left(-\frac{1}{2}\left(\left(\frac{1}{4\sigma^2+2\epsilon^2}+\frac{1}{2\epsilon^2}\right)(s_1^2+s_2^2) + 2s_1s_2\left(\frac{1}{4\sigma^2+2\epsilon^2}-\frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle\right)\right)}{(2\pi)^{n-1}(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}} s_1^{n-2}s_2^{n-2}ds_1ds_2$$

To solve this integral, let us apply the substitution $r = \sqrt{s_1/s_2}$ and $t = \sqrt{s_1 s_2}$. In this case,

$$\frac{drdt}{ds_1ds_2} = \left|\det\begin{bmatrix}\frac{\partial r}{\partial s_1} & \frac{\partial t}{\partial s_1}\\ \frac{\partial r}{\partial s_2} & \frac{\partial t}{\partial s_2}\end{bmatrix}\right| = \left|\det\begin{bmatrix}\frac{1}{2}\frac{1}{\sqrt{s_1s_2}} & \frac{1}{2}\sqrt{\frac{s_2}{s_1}}\\ -\frac{1}{2}\sqrt{\frac{s_1}{s_2^3}} & \frac{1}{2}\sqrt{\frac{s_1}{s_2}}\end{bmatrix}\right| = \frac{1}{2s_2} = \frac{r}{2t},$$

so we obtain:

$$\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^j)} = \frac{q}{A_{n-1}^2} + (1-q)\iint_{r,t>0} ...$$

$$...\frac{\exp\left(-\frac{1}{2}\left(\left(\frac{1}{4\sigma^2+2\epsilon^2}+\frac{1}{2\epsilon^2}\right)(r^2+1/r^2) + 2\left(\frac{1}{4\sigma^2+2\epsilon^2}-\frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle\right)t^2\right)}{(2\pi)^{n-1}(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}\cdot r} t^{2(n-2)+1}2drdt.$$

Using that

$$\int_0^\infty e^{-\frac{1}{2}au^2}u^{n-1}du = (2\pi/a)^{n/2}/A_n$$

we obtain:

$$\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^j)} = \frac{q}{A_{n-1}^2} + (1-q)\int_{r>0} ...$$

$$...\frac{\left(2\pi/\left(\left(\frac{1}{4\sigma^2+2\epsilon^2}+\frac{1}{2\epsilon^2}\right)(r^2+1/r^2) + 2\left(\frac{1}{4\sigma^2+2\epsilon^2}-\frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle\right)\right)^{n-1}}{(2\pi)^{n-1}(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}rA_{2(n-1)}}2dr$$

$$= \frac{q}{A_{n-1}^2} + \frac{1-q}{(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}A_{2(n-1)}} \int_{r>0} ...$$

$$...\frac{2dr/r}{\left(\left(\frac{1}{4\sigma^2+2\epsilon^2}+\frac{1}{2\epsilon^2}\right)(r^2+1/r^2)+2\left(\frac{1}{4\sigma^2+2\epsilon^2}-\frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle\right)^{n-1}}$$

Substituting $u = r^2$, we obtain:

$$\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^j)} = \frac{q}{A_{n-1}^2} + \frac{1-q}{(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}A_{2(n-1)}} \int_{u>0} ...$$

$$...\frac{du/u}{\left(\left(\frac{1}{4\sigma^2+2\epsilon^2}+\frac{1}{2\epsilon^2}\right)(u+1/u)+2\left(\frac{1}{4\sigma^2+2\epsilon^2}-\frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle\right)^{n-1}}$$

Let now:

$$a = \frac{1}{4\sigma^2+2\epsilon^2} + \frac{1}{2\epsilon^2} = \frac{\sigma^2+\epsilon^2}{(\sigma^2+\epsilon^2/2)2\epsilon^2}$$

$$b = \left(\frac{1}{4\sigma^2+2\epsilon^2} - \frac{1}{2\epsilon^2}\right)\langle x_1,x_2\rangle = -\frac{\sigma^2\langle x_1,x_2\rangle}{(\sigma^2+\epsilon^2/2)2\epsilon^2}$$

To analyze the integral

$$\int_0^\infty \frac{du/u}{(au+2b+a/u)^{n-1}}$$

we will rearrange it as:

$$\int_0^\infty \frac{du/u}{(au-2a+a/u+2b+2a)^{n-1}} =$$

$$\int_0^\infty \frac{du/u}{(a\frac{(u-1)^2}{u}+2b+2a)^{n-1}} =$$

$$\frac{1}{(2a+2b)^{n-1}} \int_0^\infty \frac{du/u}{\left(\frac{a(u-1)^2}{(2a+2b)u}+1\right)^{n-1}}$$

Now let:

$$\gamma = 1 + b/a = 1 - \frac{\langle x_1,x_2\rangle}{1+\epsilon^2/\sigma^2}$$

Note that $\gamma \geq 0$ always.

Replacing $v = (u-1)\sqrt{(n-1)/\gamma}$, we obtain:

$$\frac{1}{(2a\gamma)^{n-1}} \int_{-\sqrt{(n-1)/\gamma}}^\infty \frac{\sqrt{\gamma/(n-1)}dv/(\sqrt{\gamma/(n-1)}v+1)}{\left(\frac{v^2/(n-1)}{2\left(1+v\sqrt{\gamma/(n-1)}\right)}+1\right)^{n-1}} =$$

$$\frac{\gamma^{\frac{3}{2}-n}}{(2a)^{n-1}\sqrt{n-1}}\int_{-\sqrt{(n-1)/\gamma}}^{\infty}\frac{dv/(\sqrt{\gamma/(n-1)}v+1)}{\left(\frac{v^2/(n-1)}{2\left(1+v\sqrt{\gamma/(n-1)}\right)}+1\right)^{n-1}}$$

$$\approx \frac{\gamma^{\frac{3}{2}-n}}{(2a)^{n-1}\sqrt{n-1}}\sqrt{2\pi}$$

since the Harris/NCC feature has normally $n >> 1$ and

$$\lim_{n\to\infty}\int_{-\sqrt{(n-1)/\gamma}}^{\infty}\frac{dv/(\sqrt{\gamma/(n-1)}v+1)}{\left(\frac{v^2/(n-1)}{2\left(1+v\sqrt{\gamma/(n-1)}\right)}+1\right)^{n-1}}=\int_{-\infty}^{\infty}\frac{dv}{e^{v^2/2}}=\sqrt{2\pi}.$$

Substituting in the cost function we obtain:

$$C(x_1,x_2)\approx -\log\left(\frac{q}{A_{n-1}^2}+\frac{(1-q)\gamma^{\frac{3}{2}-n}}{(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}A_{2(n-1)}(2a)^{n-1}\sqrt{n-1}}\sqrt{2\pi}\right)$$

$$=-\log\left(\frac{q}{A_{n-1}^2}+\frac{(1-q)(\sigma^2+\epsilon^2/2)^{\frac{n-1}{2}}(2\epsilon^2)^{\frac{n-1}{2}}}{2^{n-1}(\sigma^2+\epsilon^2)^{n-1}\gamma^{n-3/2}A_{2(n-1)}\sqrt{n-1}}\sqrt{2\pi}\right)$$

$$=-\log\left(\frac{q}{A_{n-1}^2}+\frac{(1-q)}{2^{n-1}\gamma^{n-3/2}A_{2(n-1)}\sqrt{n-1}}\left(1-\frac{1}{(1+\epsilon^2/\sigma^2)^2}\right)^{\frac{n-1}{2}}\sqrt{2\pi}\right)$$

$$=-\log\left(q+\frac{(1-q)A_{n-1}^2}{2^{n-1}\gamma^{n-3/2}A_{2(n-1)}\sqrt{n-1}}\left(1-\frac{1}{(1+\epsilon^2/\sigma^2)^2}\right)^{\frac{n-1}{2}}\sqrt{2\pi}\right)+\log\left(A_{n-1}^2\right)$$

Using Equation C.3, this expression simplifies to:

$$=-\log\left(q+\frac{(1-q)A_{n-1}}{\gamma^{n-3/2}A_n\sqrt{n-1}}\left(1-\frac{1}{(1+\epsilon^2/\sigma^2)^2}\right)^{\frac{n-1}{2}}\sqrt{2\pi}\right)+\log\left(A_{n-1}^2\right)$$

which, noting from Equation C.4 that $\lim_{n\to\infty}\frac{A_{n-1}}{A_n\sqrt{n-1}}=\frac{1}{\sqrt{2\pi}}$, is approximately:

$$\approx -\log\left(q+\frac{(1-q)\left(1-\frac{1}{(1+\epsilon^2/\sigma^2)^2}\right)^{\frac{n-1}{2}}}{\left(1-\frac{\langle x_1,x_2\rangle}{1+\epsilon^2/\sigma^2}\right)^{n-3/2}}\right)+const.$$

where $const. = \log(A_{n-1}^2)$ can be disregarded, since adding a constant term (with respect to x_1 and x_2) in the cost function does not impact the final matching.

8.2 SIFT model

Instead of working directly with the SIFT feature, we will use instead the RootSIFT feature [28].

The RootSIFT approach consists in taking a SIFT feature descriptor $z \in \mathbb{R}_+^{128}$, L1-normalizing it and taking its square root componentwise, i.e. the resulting feature x_1 is written in function of z as:

$$(x_1)_i = \sqrt{z_i / ||z||_1},\ i \in \{1, ..., 128\}.$$

The logic behind this process is that Euclidean distance after taking the square root becomes equivalent to the Hellinger distance (also known as Bhattacharyya distance) prior to taking the square root, which is often a more suitable metric for comparing histograms.

Therefore x_1 satisfies:

$$||x_1||_2 = 1$$

and

$$\forall i : (x_1)_i \geq 0. \tag{8.3}$$

Note that the RootSIFT feature has therefore a unitary norm constraint, just as the Harris/NCC feature. This motivates us to use a similar model to the one we used for Harris/NCC, the difference is that it does not have the zero mean constraint. However, we will ignore the non-negativity constraint (Equation 8.3): our model assumes x_1 may have negative entries.

8.2.1 Probabilistic model

This model is almost identical to the Harris/NCC model. However, we allow anisotropic Gaussian distributions: Points in the generator set P have a variance matrix of Σ^2 and noise follows a variance matrix of E^2. Hence,

$$\text{pdf}[x] = \frac{\exp\left(-\frac{1}{2}x^T\Sigma^{-2}x\right)}{(2\pi)^{n/2}\sqrt{\det\Sigma^2}}$$

and

$$\text{pdf}[\tilde{x}_k] = \frac{\exp\left(-\frac{1}{2}\tilde{x}_k^T\tilde{\Sigma}^{-2}\tilde{x}_k\right)}{(2\pi)^{n/2}\sqrt{\det\tilde{\Sigma}^2}},\ k \in \{1, 2\}$$

where $\tilde{\Sigma}^2 = \Sigma^2 + E^2$.

The other difference from the Harris/NCC model is that only the unitary norm constraint ($||x_1|| = ||x_2|| = 1$) is used, the zero mean constraint ($\vec{1}^T x_1 = \vec{1}^T x_2 = 0$)

is not used; so the relationship between $\tilde{x}_1, \tilde{x}_2$ and x_1, x_2 is simply:

$$x_1 = \tilde{x}_1/||\tilde{x}_1||$$

$$x_2 = \tilde{x}_2/||\tilde{x}_2||$$

Note also that, because there is no zero mean constraint anymore, the measure for x_1 (volume element) becomes:

$$\frac{d\tilde{x}_1}{d\mu(x_1)ds_1} = s_1^{n-1}$$

where $\tilde{x}_1 = s_1 x_1$.

8.2.2 $dP[x_1]/d\mu(x_1)$

In the anisotropic model, the probability density function can be computed using the following integral:

$$dP[x_1]/d\mu(x_1) = \int_0^\infty \text{pdf}\left[\tilde{x}_1\right]\Big|_{\tilde{x}_1 = s x_1} s^{n-1} ds$$

$$= \int_0^\infty \frac{\exp\left(-\frac{1}{2}\left(s x_1\right)^T \tilde{\Sigma}^{-2}\left(s x_1\right)\right) s^{n-1} ds}{(2\pi)^{n/2}\sqrt{\det \tilde{\Sigma}^2}}$$

$$= \int_0^\infty \frac{\exp\left(-\frac{1}{2}\left(x_1^T \tilde{\Sigma}^{-2} x_1\right) s^2\right) s^{n-1} ds}{(2\pi)^{n/2}\sqrt{\det \tilde{\Sigma}^2}}$$

$$= \frac{1}{\sqrt{\det \tilde{\Sigma}^2}\left(x_1^T \tilde{\Sigma}^{-2} x_1\right)^{\frac{n}{2}} A_n}.$$

Note that we may rewrite this as:

$$dP[x_1]/d\mu(x_1) = \frac{1}{A_n \left(x_1^T C^{-1} x_1\right)^{\frac{n}{2}}}$$

where

$$C = \frac{\tilde{\Sigma}^2}{(\det \tilde{\Sigma}^2)^{1/n}}.$$

Also, note that C satisfies $\det C = 1$ and $C = C^T$, so it has $\frac{n(n+1)}{2} - 1$ degrees of freedom.

8.2.3 Cost function

The cost function for "max-prob" will be given by:

$$C(X_1^i, X_2^j) = -\log\left(\frac{dP[X_1^i, X_2^j|\Pi_{ij}=1]}{d\mu(X_1^i)d\mu(X_2^j)}\right)$$

$$= -\log\left(\iint_{s_1,s_2>0} F\left(s_1x_1, s_2x_2\right) s_1^{n-1}s_2^{n-1}ds_1ds_2\right)$$

where

$$F(\tilde{x}_1, \tilde{x}_2) = A(1-q) + Bq$$

$$A = \frac{\exp\left(-\frac{1}{2}\left\|\frac{\tilde{x}_1+\tilde{x}_2}{2}\right\|^2_{(\Sigma^2+E^2/2)^{-1}}\right)}{(2\pi)^{n/2}\sqrt{\det(\Sigma^2+E^2/2)}}\frac{\exp\left(-\frac{1}{2}\left\|\tilde{x}_1-\tilde{x}_2\right\|^2_{(2E^2)^{-1}}\right)}{(2\pi)^{n/2}\sqrt{\det(2E^2)}}$$

$$B = \frac{\exp\left(-\frac{1}{2}\left\|\tilde{x}_1\right\|^2_{(\Sigma^2+E^2)^{-1}}\right)}{(2\pi)^{n/2}\sqrt{\det(\Sigma^2+E^2)}}\frac{\exp\left(-\frac{1}{2}\left\|\tilde{x}_2\right\|^2_{(\Sigma^2+E^2)^{-1}}\right)}{(2\pi)^{n/2}\sqrt{\det(\Sigma^2+E^2)}}$$

We can remove the Bq term, using

$$\iint_{s_1,s_2>0} qBs_1^{n-1}s_2^{n-1}ds_1ds_2 = q\frac{dP[x_1]}{d\mu(x_1)}\frac{dP[x_2]}{d\mu(x_2)} =$$

$$\frac{q}{\det(\Sigma^2+E^2)\left(x_1^T\left(\Sigma^2+E^2\right)^{-1}x_1x_2^T\left(\Sigma^2+E^2\right)^{-1}x_2\right)^{\frac{n}{2}}A_n^2}$$

Meanwhile,

$$\iint_{s_1,s_2>0}(1-q)As_1^{n-1}s_2^{n-1}ds_1ds_2 =$$

$$\iint_{s_1,s_2>0}\frac{(1-q)e^{-\frac{1}{2}\left(\left\|s_1x_1+s_2x_2\right\|^2_{(4\Sigma^2+2E^2)^{-1}}+\left\|s_1x_1-s_2x_2\right\|^2_{(2E^2)^{-1}}\right)}}{(2\pi)^n\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}}s_1^{n-1}s_2^{n-1}ds_1ds_2$$

$$= \iint_{s_1,s_2>0}\frac{(1-q)\exp\left(-\frac{1}{2}\begin{bmatrix}s_1\\s_2\end{bmatrix}^T M\begin{bmatrix}s_1\\s_2\end{bmatrix}\right)}{(2\pi)^n\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}}s_1^{n-1}s_2^{n-1}ds_1ds_2$$

where

$$M = \begin{bmatrix}x_1^T((4\Sigma^2+2E^2)^{-1}+(2E^2)^{-1})x_1 & x_1^T((4\Sigma^2+2E^2)^{-1}-(2E^2)^{-1})x_2\\ x_2^T((4\Sigma^2+2E^2)^{-1}-(2E^2)^{-1})x_1 & x_2^T((4\Sigma^2+2E^2)^{-1}+(2E^2)^{-1})x_2\end{bmatrix}$$

Substituting $r = \sqrt{s_1/s_2}$ and $t = \sqrt{s_1 s_2}$, with $ds_1 ds_2 = 2\frac{t}{r} dr dt$, we obtain:

$$\iint_{r,t>0} \frac{(1-q)\exp\left(-\frac{1}{2}\begin{bmatrix} r \\ 1/r \end{bmatrix}^T M \begin{bmatrix} r \\ 1/r \end{bmatrix} t^2\right)}{(2\pi)^n \sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} r} t^{2(n-1)+1} 2 dr dt$$

$$= (1-q)\int_{r>0} \frac{\left(2\pi / \left(\begin{bmatrix} r \\ 1/r \end{bmatrix}^T M \begin{bmatrix} r \\ 1/r \end{bmatrix}\right)\right)^n}{(2\pi)^n \sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n} r} 2 dr$$

$$= (1-q)\int_{r>0} \frac{\left(\begin{bmatrix} r \\ 1/r \end{bmatrix}^T M \begin{bmatrix} r \\ 1/r \end{bmatrix}\right)^{-n}}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n} r} 2 dr$$

Now let

$$M = \begin{bmatrix} a & b \\ b & c \end{bmatrix},$$

substituting $u = r^2$, we obtain:

$$(1-q)\int_{u>0} \frac{(au + 2b + c/u)^{-n}}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n} u} du$$

$$= (1-q)\int_{u>0} \frac{(au - 2\sqrt{ac} + c/u + 2b + 2\sqrt{ac})^{-n}}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n} u} du$$

$$= (1-q)(2b + 2\sqrt{ac})^{-n} \int_{u>0} \frac{\left(\frac{a(u-\sqrt{c/a})^2}{u(2b+2\sqrt{ac})} + 1\right)^{-n} du/u}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n}}$$

Substituting $v = (u - \sqrt{c/a})\sqrt{\frac{n}{\gamma}}$, where $\gamma = (b + \sqrt{ac})\frac{\sqrt{c/a}}{a}$, we have then:

$$(1-q)\left(2\frac{a}{\sqrt{c/a}}\gamma\right)^{-n} \int_{-\sqrt{c/a}\sqrt{\frac{n}{\gamma}}}^{\infty} \frac{\left(\frac{av^2\gamma/n}{\left(\sqrt{c/a}+v\sqrt{\frac{\gamma}{n}}\right)2\gamma a\sqrt{\frac{a}{c}}} + 1\right)^{-n} \sqrt{\frac{\gamma}{n}} dv / \left(\sqrt{\frac{c}{a}} + v\sqrt{\frac{\gamma}{n}}\right)}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n}}$$

$$= (1-q)\left(2\frac{a}{\sqrt{c/a}}\gamma\right)^{-n} \frac{\sqrt{\frac{\gamma}{n}}}{\sqrt{c/a}} \int_{-\sqrt{c/a}\sqrt{\frac{n}{\gamma}}}^{\infty} \frac{\left(\frac{v^2/n}{2\left(1+v\sqrt{\frac{a}{c}}\sqrt{\frac{\gamma}{n}}\right)} + 1\right)^{-n} dv / \left(1 + v\sqrt{\frac{a}{c}}\sqrt{\frac{\gamma}{n}}\right)}{\sqrt{\det(\Sigma^2 + E^2/2)}\sqrt{\det(2E^2)} A_{2n}}.$$

As $n \to \infty$, the result

$$\lim_{n\to\infty} \int_{-\sqrt{c/a}\sqrt{\frac{n}{\gamma}}}^{\infty} \frac{\left(\frac{v^2/n}{2\left(1+v\sqrt{\frac{a}{c}}\sqrt{\frac{\gamma}{n}}\right)}+1\right)^{-n} dv}{\left(1+v\sqrt{\frac{a}{c}}\sqrt{\frac{\gamma}{n}}\right)} = \int_{-\infty}^{\infty} e^{-v^2/2} = \sqrt{2\pi}$$

lets us approximate the expression for the "A" term to:

$$\frac{(1-q)\left(2\frac{a}{\sqrt{c/a}}\gamma\right)^{-n}\sqrt{\frac{\gamma}{n}}}{\sqrt{c/a}} \frac{\sqrt{2\pi}}{\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}A_{2n}}$$

$$= \frac{(1-q)2^{-n}a^{-n}\left(\frac{\gamma}{\sqrt{c/a}}\right)^{-(n-1/2)} \frac{1}{(\sqrt{c/a})^{1/2}}\sqrt{2\pi}}{\sqrt{n}\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}A_{2n}}$$

$$= \frac{(1-q)2^{-n}\frac{1}{\sqrt{a}}\left(\frac{a\gamma}{\sqrt{c/a}}\right)^{-(n-1/2)} \frac{1}{(\sqrt{c/a})^{1/2}}\sqrt{2\pi}}{\sqrt{n}\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}A_{2n}}$$

$$= \frac{(1-q)2^{-n}\left(b+\sqrt{ac}\right)^{-(n-1/2)} \frac{1}{(ac)^{1/4}}\sqrt{2\pi}}{\sqrt{n}\sqrt{\det(\Sigma^2+E^2/2)}\sqrt{\det(2E^2)}A_{2n}}$$

Using that[2]:

$$\lim_{n\to\infty} \frac{A_n^2}{2^n A_{2n}\sqrt{n}} = \frac{1}{\sqrt{2\pi}}$$

we can approximate the expression to

$$\frac{(1-q)\left(b+\sqrt{ac}\right)^{-(n-1/2)} \frac{1}{(ac)^{1/4}}}{\sqrt{\det(2E^2)}\sqrt{\det(\Sigma^2+E^2/2)A_n^2}}$$

so the final expression for cost is:

$$C(x_1,x_2) \approx -\log\left(\tilde{A}+\tilde{B}\right) + \log(A_{n-1}^2)$$

where

$$\tilde{A} = \frac{(1-q)\left(b+\sqrt{ac}\right)^{-(n-1/2)} \frac{1}{(ac)^{1/4}}}{\sqrt{\det(2E^2)}\sqrt{\det(\Sigma^2+E^2/2)}}$$

$$a = x_1^T\left[(4\Sigma^2+2E^2)^{-1}+(2E^2)^{-1}\right]x_1$$

$$b = x_1^T\left[(4\Sigma^2+2E^2)^{-1}-(2E^2)^{-1}\right]x_2$$

$$c = x_2^T\left[(4\Sigma^2+2E^2)^{-1}+(2E^2)^{-1}\right]x_2$$

[2]See Appendix C.

$$\tilde{B} = \frac{q}{\det(\Sigma^2 + E^2)\left(x_1^T (\Sigma^2 + E^2)^{-1} x_1 x_2^T (\Sigma^2 + E^2)^{-1} x_2\right)^{\frac{n}{2}}}$$

and the $\log(A_{n-1}^2)$ term may be disregarded.

Note that, just as the Harris/NCC model depended only on ϵ/σ, and not on ϵ and σ individually, similarly, our RootSIFT model is invariant to multiplying Σ^2 and E^2 by a same scale α.

8.2.4 Maximum Likelihood Estimation

Computing the cost function requires knowledge of Σ^2 and E^2. While acquiring both of them is no trivial task, it is possible to infer the value of $C = \frac{\Sigma^2+E^2}{\det(\Sigma^2+E^2)^{1/n}}$ from a set of features using a maximum likelihood estimator (MLE) method.

Given a set of features $\{X^1, X^2, ..., X^{\#\text{samples}}\} \subset \mathbb{R}^n$, the MLE method in our case must seek a symmetric positive definite matrix C, satisfying $\det C = 1$, that maximizes:

$$\prod_{i=1}^{\#\text{samples}} \frac{dP[X^i|C]}{d\mu(X^i)} = \prod_{i=1}^{\#\text{samples}} \frac{1}{((X^i)^T C^{-1} X^i)^{\frac{n}{2}} A_n}.$$

Derivative

Let us first remove the constraint that $\det C = 1$ and write instead

$$\frac{dP[X^i|C]}{d\mu(X^i)} = \frac{1}{\sqrt{\det C}\,((X^i)^T C^{-1} X^i)^{\frac{n}{2}} A_n}$$

The MLE method maximizes:

$$\begin{aligned} f(C) &= \sum_{i=1}^{\#\text{samples}} \log \frac{dP[X^i|C]}{d\mu(X^i)} \\ &= \sum_{i=1}^{\#\text{samples}} -\frac{1}{2}\log(\det C) - \log(A_n) - \frac{n}{2}\log\left((X^i)^T C^{-1} X^i\right) \\ &= \sum_{i=1}^{\#\text{samples}} -\frac{1}{2}\log(\det C) - \log(A_n) - \frac{n}{2}\log\left(C^{-1} : (X^i (X^i)^T)\right) \end{aligned}$$

Let us compute the gradient of this function, i.e. a matrix $\partial_C f(C)$ such that:

$$f(C + \delta C) = f(C) + \partial_C f : \delta C + o(||\delta C||_F), \text{ (as } ||\delta C||_F \to 0)$$

where $||\cdot||_F$ is the Frobenius norm and $A : B$ denotes matrix inner-product ($A : B = \sum_{i,j} A_{ij} B_{ij}$). In other words, $(\partial_C f(C))_{ij} = \partial_{C_{ij}} f(C)$.

For that purpose we will make use of the following matrix derivatives:

$$A : (B + \delta B) = A : B + A : \delta B \Rightarrow \partial_B (A : B) = A; \tag{8.4}$$

$$
\begin{aligned}
(A+\delta A)^{-1} &= A^{-1}(I+\delta A A^{-1})^{-1} = A^{-1}\sum_{i=0}^{\infty}(-\delta A A^{-1})^i \\
&= A^{-1} - A^{-1}\delta A A^{-1} + o(||\delta A||_F) \Rightarrow \\
\partial_A(A^{-1}:B):\delta A &= -(A^{-1}\delta A A^{-1}):B = -\delta A : A^{-T}BA^{-T} \Rightarrow \\
\partial_A(A^{-1}:B) &= -A^{-T}BA^{-T}; \qquad (8.5)
\end{aligned}
$$

$$
\begin{aligned}
\det(A+\delta A) &= \det(A)\cdot\det(I+A^{-1}\delta A) \\
&= \det(A) + \det(A)\cdot\operatorname{tr}(A^{-1}\delta A) + o(||\delta A||_F) \\
&= \det(A) + \det(A)\cdot(A^{-T}:\delta A) + o(||\delta A||_F) \Rightarrow \\
\partial_A(\det A) &= (\det A)\cdot A^{-T}; \qquad (8.6)
\end{aligned}
$$

resulting in:

$$
\begin{aligned}
\partial_C f(C) &= \sum_{i=1}^{\#\text{samples}} -\frac{1}{2}C^{-T} + \frac{n}{2}\frac{C^{-T}X^i(X^i)^TC^{-T}}{C^{-1}:(X^i(X^i)^T)} \\
&= \sum_{i=1}^{\#\text{samples}} -\frac{1}{2}C^{-1} + \frac{n}{2}\frac{C^{-1}X^i(X^i)^TC^{-1}}{(X^i)^TC^{-1}X^i}.
\end{aligned}
$$

Iterative Method

The maximum occurs when the gradient is zero, which means:

$$
\sum_{i=1}^{\#\text{samples}} -\frac{1}{2}C^{-1} + \frac{n}{2}\frac{C^{-1}X^i(X^i)^TC^{-1}}{(X^i)^TC^{-1}X^i} = 0 \Rightarrow
$$

$$
C^{-1} = \frac{n}{\#\text{samples}}\sum_{i=1}^{\#\text{samples}}\frac{C^{-1}X^i(X^i)^TC^{-1}}{(X^i)^TC^{-1}X^i} \Rightarrow \qquad (8.7)
$$

$$
C = \frac{n}{\#\text{samples}}\sum_{i=1}^{\#\text{samples}}\frac{X^i(X^i)^T}{(X^i)^TC^{-1}X^i}. \qquad (8.8)
$$

Notice that our final equation for C has the term "C" on both sides in a non-separable way. One may use Equation 8.8 as an iteration, which would be equivalent to updating C according to a preconditioned gradient ascent method (following the step $\delta C = \frac{2C(\partial_C f)C}{\#\text{samples}}$), starting for instance with $C = I$. Additionally, one may want to normalize $C := C/(\det C)^{1/n}$ after each iteration for stability, which will ensure that $\det C = 1$.

Alternatively, one may want to use Equation 8.7 for the iterative method, i.e. working directly on the inverse C^{-1} instead. However, this would not work since it is equivalent to walking on the decreasing direction of the gradient, not the increasing one.

We experimentally verified that the method that iterates on Equation 8.8 converges very fast, often in less than 20 iterations.

Chapter 9

Evaluation in Computer Vision

In this chapter we evaluate our models for the Harris/NCC and RootSIFT features using Mikolajczyk's dataset[1] [16].

9.1 Methodology

Mikolajczyk's dataset is a small dataset that provides images taken from a same scene distorted in different forms. In this work, we use the "graf", "bikes", "wall" and "trees" subsets (Figure 9.1). While the "graf" and "wall" subsets explore noise caused by change of viewpoint, "bikes" and "trees" explore noise caused by blur. The dataset also provides the homographies that relate the 1st and the n-th image of each subset ($n = \{2, 3, ..., 6\}$).

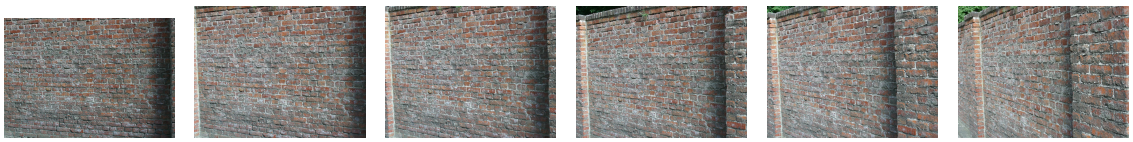
We detect features[2] in each image and match the feature sets using several methods. The ground truth is determined using the homography provided by Mikolajczyk's dataset and the 2D location of each feature: a match pair (x_1, x_2), with image coordinates (l_1, l_2), is considered correct if the projection of l_1 in the other image, following the homography, has an Euclidean distance of less than 4 pixels from l_2. l_1 always refers to a point in the 1st image, while l_2 refers to a point in the n-th image ($n \in \{2, ..., 6\}$), of each subset of the dataset. Note also that this ground-truth criterion allows many-to-many matching, even if we only consider one-to-one matching methods.

[1] Available at `http://www.robots.ox.ac.uk/~vgg/research/affine/`

[2] For Harris we used a custom implementation, while for SIFT we used Lowe's software available at `http://www.cs.ubc.ca/~lowe/keypoints/`. To reduce the number of SIFT features, we discarded the features of the 2 lowest scales of the images in the "graf" and "bikes" subsets and the 4 lowest scales in the "wall" and "trees" subsets. Our NCC descriptor used 21×21 patches with 3 color channels, so that $n = 1323$, while the SIFT descriptor is 128-dimensional, as default, using only grayscale information.

(a) graf-1 (b) graf-2 (c) graf-3 (d) graf-4 (e) graf-5 (f) graf-6

(g) bikes-1 (h) bikes-2 (i) bikes-3 (j) bikes-4 (k) bikes-5 (l) bikes-6



(m) wall-1 (n) wall-2 (o) wall-3 (p) wall-4 (q) wall-5 (r) wall-6

(s) trees-1 (t) trees-2 (u) trees-3 (v) trees-4 (w) trees-5 (x) trees-6

Figure 9.1: Images from Mikolajczyk's dataset used.

9.1.1 Methods

We evaluate the following methods (in parenthesis is the acronym we will use to refer to each method in this chapter):

- Greedy #2, using Euclidean distance as cost **(G2)**
- Minimum bipartite matching with Euclidean distance **(E1)**
- Minimum bipartite matching with squared Euclidean distance **(E2)**
- "max-prob" with isotropic Gaussian distributions and outliers **(GA)**
- Our Harris/NCC model **(HN)**
- Our RootSIFT model with isotropic distributions, i.e. $\Sigma^2 = \sigma^2 I$ and $E^2 = \epsilon^2 I$ **(II)**
- Our RootSIFT model with an anisotropic generator set distribution, but isotropic noise **(AI)**
- Our RootSIFT model with anisotropic generator set distribution and anisotropic noise **(AA)**

While G2, E2 and E1 have no parameters, the other methods require knowing parameters such as the noise ratio and the outlier rate.

GA has three parameters: σ, ϵ and q. Because Harris/NCC and RootSIFT features satisfy $||x_1|| = 1$, meaning that $E[||x_1||^2] = 1$, we force the Gaussian model to satisfy $E[||x_1||^2] = 1$ by constraining $\sigma^2 + \epsilon^2 = 1/n$. We then choose ϵ and σ that satisfy $\frac{\epsilon^2}{\sigma^2+\epsilon^2} = \chi^2$, for an input parameter[3] $\chi \in]0, 1[$. Therefore GA has two parameters: q and χ.

The HN and II methods also have two parameters: q and the ratio ϵ/σ. Similarly to the GA method, we choose ϵ/σ satisfying $\frac{\epsilon^2}{\sigma^2+\epsilon^2} = \chi^2$, for given $\chi \in]0, 1[$.

The AI method requires knowing the variance matrix Σ^2. For this end, we use the MLE method to estimate $\tilde{\Sigma}^2 = \Sigma^2 + \epsilon^2 I$, requiring that $\tilde{\Sigma}^2$ is a diagonal matrix[4] (we modify the MLE method seen in Section 8.2.4 to remove the non-diagonal components of C after each iteration). $\tilde{\Sigma}^2$ is estimated using the input features $(P_1 \cup P_2)$ of each test case. After computing $\tilde{\Sigma}^2$, we set $\epsilon^2 = \chi^2 \min_i(\tilde{\Sigma}^2)_{ii}$ and do $\Sigma^2 = \tilde{\Sigma}^2 - \epsilon^2 I$.

[3] I.e., (σ, ϵ) is the solution to the system $\begin{cases} \sigma^2+\epsilon^2=1/n \\ \epsilon^2/(\sigma^2+\epsilon^2)=\chi^2 \end{cases}$.

[4] The choice of using a diagonal variance matrix for the SIFT case is reasonable because of the very nature of this feature model: The SIFT feature descriptor is built aligned with the orientation (rotation) with the highest occurrence of image gradients; while the descriptor *is* a histogram of gradients that stores gradients according to orientation. Therefore, it is natural that some components of the resulting RootSIFT vector tend to have higher values than other components, due to this alignment.

The AA method uses the same MLE method as AI, but the noise matrix is estimated as $E^2 = \chi^2\tilde{\Sigma}^2$, and $\Sigma^2 = \tilde{\Sigma}^2 - E^2$.

This way all methods depend only on two parameters: χ and q, except for G2, E2 and E1, which have no parameters.

9.1.2 Parameter selection

A preliminary experiment shows us that the methods we are evaluating are very robust to changes in q: Figure 9.2 shows that changing q makes no difference at all in most cases — the curves for each value of q overlap almost entirely each other. Therefore, in the next experiments we set $q = .5$ and vary only $\chi \in \{.02, .04, .06, ..., .98\}$.

9.2 Results

We vary $\chi \in \{.02, .04, .06, ..., .98\}$ and select the maximum and median hit counts obtained for each method in order to compare the methods to each other. The maximum hit count should evaluate how many correct matches we can obtain supposing we know the correct noise ratio χ, while the median hit count should roughly evaluate how many correct matches we have when we simply guess χ.

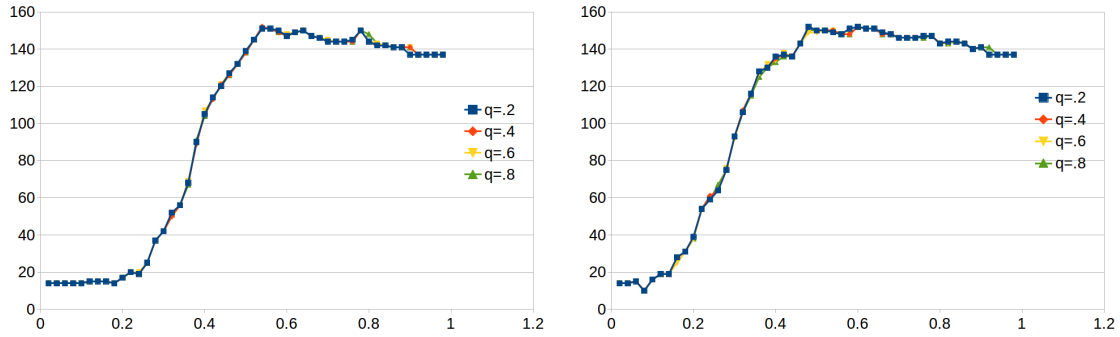
In Table 9.1 we see the results for Harris/NCC features, while in Table 9.2 we see the results for RootSIFT[5] features. For the parametric methods, we use the notation "MAX|MED", where MAX is the maximum hit count and MED is the median. We also enhance in boldface, for each test set, the parameter-less method with the highest hit count, the parametric method with the highest maximum hit count, and the parametric method with the highest median hit count. In the bottom row we sum the number of times each method was displayed in boldface, i.e. the number of times it outperformed the other methods of the same category.

From the tables we see that G2 and E1 had better hit counts than E2 in general. Because E2 is equivalent to the Gaussian method with $q = 0$, it is sensitive to outliers, while G2's greedy nature tends to prioritize inliers, and E1 tends to give a lower cost to outlier pairs, since it does not square the distance between the pairs as E2 does.

For the Harris/NCC parametric methods, we see that GA and HN have similar maximum hit counts (only slightly higher for HN), while HN has a much better median hit count. This means that it is easier to select χ for HN than for GA.

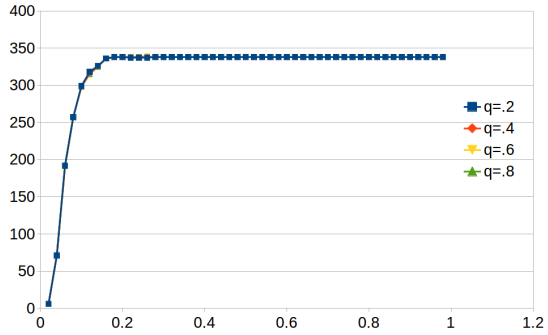
For RootSIFT, the AI method had the best results, while AA had the worst results among parametric methods. This suggests that the noise distribution is

[5]The parameter-less and Gaussian methods also use the RootSIFT feature instead of SIFT.

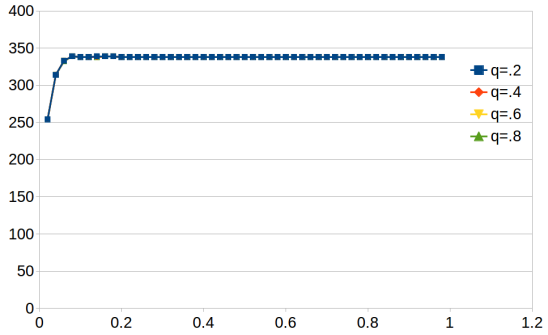


(a) Harris/NCC GA

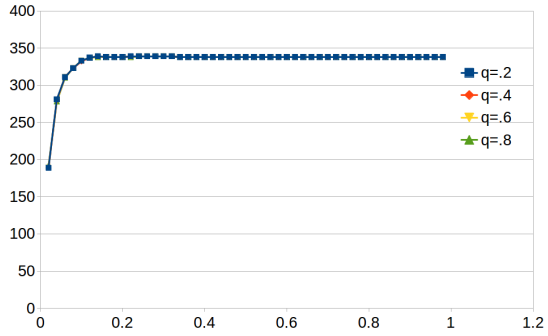
(b) Harris/NCC HN



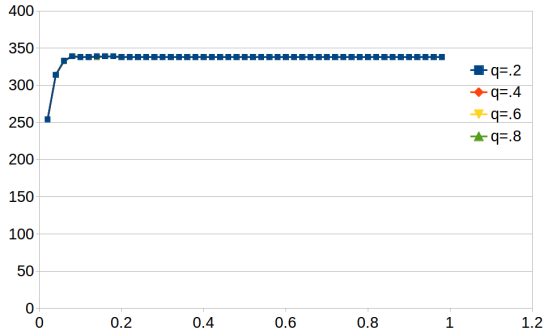
(c) RootSIFT GA



(d) RootSIFT II



(e) RootSIFT AI



(f) RootSIFT AA

Figure 9.2: Varying q and χ for different methods (graf1-2 case). In the x-axis is the value of χ and in the y-axis is the hit count.

Table 9.1: Hit count comparison for Harris/NCC features

case	#features	G2	E2	E1	GA	HN
graf1-2	478×488	128	137	**148**	**152**|**137**	**152**|**137**
graf1-3	478×483	**71**	69	70	**71**|63	**71**|**66**
graf1-4	478×482	**10**	**10**	**10**	**11**|**8**	**11**|**8**
graf1-5	478×484	22	**31**	30	**33**|27	31|**28**
graf1-6	478×468	7	**8**	6	**10**|4	9|**7**
bikes1-2	483×495	329	338	**344**	**343**|338	**343**|**339**
bikes1-3	483×489	301	308	**311**	310|308	**313**|**311**
bikes1-4	483×489	221	230	**236**	235|229	**236**|**232**
bikes1-5	483×485	143	149	**155**	157|144	**163**|**149**
bikes1-6	483×482	67	**80**	76	83|76	**86**|**77**
wall1-2	480×490	**337**	334	336	**337**|331	**337**|**334**
wall1-3	480×483	**298**	297	297	**301**|281	300|**292**
wall1-4	480×478	**194**	192	**194**	**195**|170	**195**|**185**
wall1-5	480×487	113	121	**128**	**127**|83	125|**106**
wall1-6	480×492	34	41	**42**	**42**|17	**42**|**23**
trees1-2	487×482	**210**	206	207	208|199	**209**|**206**
trees1-3	487×488	**168**	167	**168**	167|164	**169**|**166**
trees1-4	487×489	74	76	**77**	**77**|73	**77**|**74**
trees1-5	487×475	44	**47**	45	**49**|**45**	48|**45**
trees1-6	487×486	15	**18**	17	19|16	**21**|**17**
	bold count	7	6	11	13|3	15|20

Table 9.2: Hit count comparison for RootSIFT features. Note: The "case" column abbreviates "graf", "bikes", "wall" and "trees" respectively as "G", "B", "W" and "T"

case	#features	G2	E2	E1	GA	II	AI	AA
G1-2	636×742	**341**	338	338	338|**338**	**339**|**338**	**339**|**338**	338|337
G1-3	636×885	**211**	207	206	212|207	212|207	**213**|**209**	210|203
G1-4	636×909	**76**	74	**76**	79|74	**80**|**75**	79|74	75|73
G1-5	636×1009	**19**	**19**	**19**	19|**19**	19|**19**	**21**|18	15|13
G1-6	636×1120	7	7	**8**	7|**7**	8|**7**	**9**|**7**	6|6
B1-2	653×428	310	**313**	**313**	313|313	**314**|313	313|312	**314**|**314**
B1-3	653×268	**206**	**206**	**206**	**206**|**206**	**206**|**206**	**206**|**206**	**206**|**206**
B1-4	653×143	**105**	**105**	**105**	**105**|**105**	**105**|**105**	**105**|**105**	**105**|**105**
B1-5	653×102	**68**	**68**	**68**	**68**|**68**	**68**|**68**	**68**|**68**	**68**|**68**
B1-6	653×68	**50**	49	**50**	**50**|49	**50**|49	**50**|**50**	**50**|**50**
W-2	514×650	**288**	286	286	286|**286**	287|**286**	**288**|**286**	286|285
W1-3	514×635	**215**	214	**215**	**215**|214	**215**|**215**	**215**|**215**	214|214
W1-4	514×612	136	135	**137**	**137**|135	**137**|**136**	**137**|**136**	**137**|**136**
W1-5	514×657	**90**	83	83	**90**|83	**90**|**84**	**90**|**84**	89|83
W1-6	514×629	**19**	**19**	**19**	20|**19**	20|**19**	**22**|**19**	17|16
T1-2	797×742	**289**	287	287	289|**287**	289|**287**	289|**287**	**290**|**287**
T1-3	797×934	297	297	**300**	298|297	**300**|297	**300**|**299**	297|295
T1-4	797×700	**192**	188	188	**195**|**188**	194|**188**	**195**|**188**	**195**|**188**
T1-5	797×361	**103**	101	100	102|101	**103**|101	**103**|**102**	102|101
T1-6	797×227	60	**61**	**61**	61|61	61|61	61|61	**62**|**62**
	bold count	15	7	13	8|10	12|14	16|16	9|9

most likely not aligned with the generator set distribution for SIFT features. On the other hand, it also suggests that a diagonal variance matrix is suitable[6] to model this type of feature, even if we are considering only isotropic noise.

When we compare the parametric methods with the parameter-less methods, we note that, while parametric methods tend to outperform the parameter-less methods when maximum hit count is considered, parameter-less methods have better results when it comes to median hit count. This means that, if we know the correct noise ratio χ we will outperform G2 and E1 using HN or AI, but when we have no clue of the correct χ it is better to choose G2 or E1 instead.

One may question if it is even possible to have any means to know the best value of χ in advance, as the maximum argument of the hit count in function of χ is hardly salient (we can see this in Figure 9.2), i.e. comparing the maximum hit count of a parametric method with the hit count of a parameter-less method is rather unfair. Most likely, methods such HN and AI are just as good as G2 and E1, but if we look only to the maximum hit count, our conclusion would be biased towards the parametric methods.

Interestingly, we see that while Harris/NCC methods have some discrepancy in hit count, RootSIFT methods produce very similar results to each other. There are even cases, namely "bikes1-3", "bikes1-4" and "bikes1-5", where *all* methods have exactly the same hit count, including maximum and median hit counts. Figure 9.2 also shows how RootSIFT is much more robust to changes in χ than Harris/NCC. This is probably because the SIFT feature descriptor was designed to be resistant to the main sources of noise in images, particularly rotation and isotropic scale. As a result, inlier pairs tend to be very close to each other, so that no much effort is required in order to match pairs correctly.

Finally, while there is some difference in hit count between the different methods, if we compare with the total number of features, the difference is indeed minute. Considering that the matching method is succeeded by a RANSAC-like procedure [23] to remove outliers and solve the application in question, e.g. 3D reconstruction or image stitching, this change in hit count does as good as no benefit to the output of the final application. Therefore in practice, for this sort of application, using a greedy algorithm is a better option due to its simplicity and lower computational cost.

[6] i.e., more suitable than isotropic variance, which is not true for the Harris/NCC feature

Chapter 10

Conclusion

We presented a probabilistic framework for matching problems, from which we could derive optimal Bayesian methods and asymptotic properties. We also instantiated it in the feature matching problem of computer vision and compared it to existing approaches.

We have learned that there is a fundamental relationship between the amount of noise ϵ, the number of dimensions n and the number of points N, where $\epsilon^n = o(1/N)$ guarantees 100% hit rate, and $\epsilon^n = o(1/N^2)$ guarantees 100% probability of matching all pairs correctly.

We learned that different distributions may have very different asymptotic behaviors: For instance, Gaussian distributions in the generator set model have a hit count of $(1 + \sigma^2/\epsilon^2)^n$ as $N \to \infty$ using our "max-prob" method, while power law and exponential distributions have an infinite hit count — the exponential distribution has a logarithmically growing hit count while the hit count of the power law distribution grows following a power law.

In the computer vision study case, our methods did not substantially improve the hit count for feature matching, but were not worse than the existing approaches either.

10.1 Future work

Many unanswered questions and unexplored subproblems remain to be studied, namely:

- It would be interesting to explore other applications for our framework, possibly out of the computer vision field.
- Methods that remove outliers, such as 2-NN (Section 2.1.3) or the method proposed in Section 4.3.2, deserve further studies. While they tend to have a higher hit rate but a lower hit count compared to Greedy #2 for instance, both

metrics are important when it comes to post-processing using RANSAC [23] for instance: While a higher hit count improves the final result of the application, a higher hit rate diminishes the computational cost of RANSAC.

- Although our lower bounds suppose that the number of dimensions n is fixed, they suggest that matching is easier when the number of dimensions is higher. However, many methods, such as PCA-SIFT [29], use dimensionality reduction in their favor. Therefore it would be interesting to study also the asymptotics of the number of dimensions n and its implications.

- Another problem that deserves further studies is *probabilistic point querying* (Appendix F), which may have implications in recognition (classification) problems in artificial intelligence.

- Deriving upper bounds, to show that some of our lower bounds are tight (e.g. showing that $\epsilon^n = o(1/N)$ is not only a sufficient condition, but also a necessary condition to guarantee 100% hit rate);

- Intra-set correlation models (i.e., the idea that two points of the same set, $X_1^i, X_1^j \in P_1$, may be correlated, differently from what our framework models) may be useful in many computer vision applications, possibly using graph matching techniques. Presenting probabilistic models that capture this sort of correlation is something that was not explored in this work and yet may produce powerful methods.

Appendices

Appendix A

List of symbols and notation

- $P[x]$, pdf$[x]$, $E[x]$: Respectively probability (or probability mass), probability density and expectation of a random variable x. Note: we do not use the convention of writing random variables with capital letters, a symbol should be identified as constant or random by context.
- P: generator set $P \subset \mathbb{R}^n$.
- P_1, P_2: Observed sets. $P_1 = P$ when the direct model is being used.
- N: Number of points in P, P_1, P_2.
- n: number of dimensions
- X, X_1, X_2: Sets P, P_1 and P_2 in matrix form $(n \times N)$.
- x_1, x_2: A point from P_1 and a point from P_2. They may or may not be generated from a same point $x \in P$, and they may or may not be inliers, depending on the context.
- x_2^*: The point $x_2^* \in P_2$ that the "max-prob" method chooses as a match to $x_1 \in P_1$.
- X_1^i or X_2^i: i-th column of matrix X_1 or X_2.
- Y_1, Y_2: Noise in matrix form $(n \times N)$. When the direct model is used, simply Y.
- p, p_1, p_2: Probability density of the points in P, P_1 and P_2. In the direct model, $p = p_1 \neq p_2$, in the generator set model, $p \neq p_1 = p_2$. In Gaussian isotropic models, normally $p(x) = g_\sigma(x)$ and $p_2(x_2) = g_{\sqrt{\sigma^2+\epsilon^2}}(x_2)$.
- p_y: Noise distribution (probability density). In a Gaussian model, normally $p_y(y) = g_\epsilon(y)$.

- $f * g$: Convolution in $\mathbb{R}^n$: $\{f * g\}(x) = \int_{\mathbb{R}^n} f(y)g(x-y)dy$. Additionally, $g_\sigma * g_\epsilon = g_{\sqrt{\sigma^2+\epsilon^2}}$ for any σ, ϵ.
- *isotropic* distribution: In this work we say that a random variable $X \in \mathbb{R}^n$ has an isotropic distribution if it satisfies $\text{pdf}[X] = f(||X||)$, for some $f : \mathbb{R} \to \mathbb{R}$.
- $g_\sigma(x)$: n-dimensional Gaussian (normal) probability density function with parameter σ (isotropic). $g_\sigma(x) = e^{-\frac{||x||^2}{2\sigma^2}}/(2\pi\sigma^2)^{n/2}$. The anisotropic Gaussian distribution takes a covariance matrix Σ^2 and has the form $p(x) = e^{-\frac{x^T\Sigma^{-2}x}{2}}/\sqrt{(2\pi)^n \det \Sigma^2}$.
- σ: parameter of the generator set distribution in the Gaussian model (isotropic).
- ϵ: parameter of the noise distribution in the Gaussian model (isotropic).
- Σ^2 and E^2: Matrix counterparts of σ^2 and ϵ^2 when anisotropic distributions are considered. Here, "M^2" is an abuse of notation for "MM^T".
- A^{-T}, for any matrix A: Inverse transpose. $A^{-T} = (A^{-1})^T = (A^T)^{-1}$.
- D: difference between x_1 and x_2 (inliers): $D = x_1 - x_2$. Used only in the generator set model with Gaussian noise. $\text{pdf}[D] = g_{\sqrt{2}\epsilon}(D)$ for any generator set distribution.
- M: mean between a pair x_1 and x_2 (inliers): $M = (x_1 + x_2)/2$. Used only in the generator set model with Gaussian noise. Has a probability density of $p_m(M) = \{p * g_{\epsilon/\sqrt{2}}\}(M)$. If the generator set is also Gaussian, then $p_m = g_\sigma * g_{\epsilon/\sqrt{2}} = g_{\sqrt{\sigma^2+\epsilon^2/2}}$.
- p_m: probability density of M.
- Π: Permutation matrix ($N \times N$). In the variational formulation, it represents a functional $\Pi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$
- π: Permutation function. Usually $\pi(i) = j \Leftrightarrow \Pi_{ij} = 1$.
- u, v: The dual variables of the minimum bipartite matching problem (vertex labelings). They are vectors in $\mathbb{R}^N$ in the finite version (fixed N), and functionals $u, v : \mathbb{R}^n \to \mathbb{R}$ in the variational version ($N \to \infty$).
- S: "selection" matrix with $S_{i,i} = 0$ with probability q (or $q' = 1-\sqrt{1-q}$ in the symmetric outlier model) and $S_{i,i} = 1$ with probability $1-q$ (or $1-q' = \sqrt{1-q}$ in the symmetric outlier model), and $S_{ij} = 0$ for all $i \neq j$. In the symmetric outlier model, it is separated in two matrices S_1 and S_2. S may also refer to the match set returned by the matching algorithm.

- $\text{Per}(A)$: Permanent of a matrix. By definition $\text{Per}(A) = \sum_{\pi} \prod_{i=1}^{N} A_{i,\pi(i)}$. Despite the similarity with the determinant definition ($\det(A) = \sum_{\pi} \text{sgn}(\pi) \prod_{i=1}^{N} A_{i,\pi(i)}$), computing the permanent is no easy task: The fastest known exact algorithm to compute it is $O(2^N N)$ [25].

- $\langle x, y \rangle$: Inner product ($x^T y$)

- $\langle x, y \rangle_S$: S-norm inner product ($x^T S y$)

- $||x||_S^2$: Denotes $x^T S x$ for some symmetric positive definite matrix S.

- $||A||_F^2$: Frobenius norm of a matrix: $||A||_F^2 = A : A$.

- $||x||_k$, for any $k = \{1, 2, ...\}$: L^k norm of x; i.e., $||x||_k = \left(\sum_{i=1}^{n} x_i^k\right)^{1/k}$, where x_i denotes the i-th component of x.

- $A : B$: Matrix inner product: $A : B = \sum_{i,j} A_{ij} B_{ij}$. Satisfies $(AB) : C = A : (CB^T) = B : (A^T C)$.

- R_{*ij}: The $(N-1) \times (N-1)$ matrix obtained after removing line i and row j from the $N \times N$ matrix R.

- $h(x_1, x_2)$: joint probability density of x_1 and x_2, given that both come from the same point $x \in P$ and are inliers. $h(x) = \text{pdf}[x_1, x_2] = \int_{\mathbb{R}^n} \text{pdf}[x_1|x]\text{pdf}[x_2|x]\text{pdf}[x]dx$.

- $H(x_1, x_2)$: Normalized joint probability #1. $H(x_1, x_2) \triangleq \frac{h(x_1,x_2)}{\sqrt{h(x_1,x_1)h(x_2,x_2)}}$. Useful when a generator set model is being used, because it is non-negative everywhere.

- $\zeta(x_1, x_2)$: Normalized joint probability #2. $\zeta(x_1, x_2) \triangleq \frac{h(x_1,x_2)}{p_1(x_1)p_2(x_2)}$. Useful when matching sets of different sizes; also appears in a number of threshold criteria.

- $\tilde{h}(x_1, x_2)$, $\tilde{H}(x_1, x_2)$ and $\tilde{\zeta}(x_1, x_2)$: The same as h, H and ζ, but taking into account the possibility of outliers.

- $C, \tilde{C}$: cost matrix of the "max-prob" and "max-expect" methods ($N \times N$). Also written $C(x_1, x_2)$ in the variational version, here a function $C : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

- A_n: See Appendix C

- $A \wedge B$: A and B (logical conjunction). Similarly, $\bigwedge_{i=1}^{N} A_i$ denotes $A_1 \wedge ... \wedge A_n$.

- $O(f(x))$: See Appendix B

- $o(f(x))$: See Appendix B
- $\omega(f(x))$: See Appendix B
- $\Omega(f(x))$: See Appendix B
- $\Theta(f(x))$: See Appendix B
- $f \sim g$: See Appendix B
- $f(x) \gtrsim g(x)$: Denotes $f(x) \geq \tilde{g}(x)$ for some $\tilde{g}$ such that $\tilde{g}(x) \sim g(x)$. I.e., if for instance $x \to \infty$, this is equivalent to $(\forall \gamma > 0)(\exists \bar{x}) : (x > \bar{x}) \Rightarrow f(x) \geq (1-\gamma)g(x)$, or equivalently, $\liminf_{x\to\infty} \frac{f(x)}{g(x)} \geq 1$. The "$\lesssim$" symbol is analogous.
- Q, $\bar{Q}$: hit rate (or its lower bound)
- $\bar{r}$: safety radius (usually for restricting $||D|| \leq \epsilon\bar{r}$).
- λ: scale parameter of the exponential distribution ($\text{pdf}[x] \propto e^{-\lambda||x||}$)
- α: shape parameter of the power law distribution ($\text{pdf}[x] \propto ||x/m||^{-\alpha}$)
- m: scale parameter of the power law distribution ($\text{pdf}[x] \propto ||x/m||^{-\alpha}$)
- $\triangleq$: equal by definition
- $\vec{1}$: The vector $\left[\begin{smallmatrix}1\\ \cdots \\ 1\end{smallmatrix}\right]$. Also written $\vec{1}_k$, where k is the number of dimensions.

Appendix B

Complexity

In this dissertation we employ the symbols $O(\cdot)$, $\Omega(\cdot)$, $o(\cdot)$, $\omega(\cdot)$, $\Theta(\cdot)$, and $\sim$ to describe asymptotic behavior of functions.

For a variable $x \to \infty$, the definitions of $O(\cdot)$, $\Omega(\cdot)$, $o(\cdot)$ and $\omega(\cdot)$ are:

$$f(x) = O(g(x)) \Leftrightarrow (\exists \gamma > 0)(\exists x_0 > 0) : x > x_0 \Rightarrow f(x) < \gamma g(x)$$

$$f(x) = o(g(x)) \Leftrightarrow (\forall \gamma > 0)(\exists x_0 > 0) : x > x_0 \Rightarrow f(x) < \gamma g(x)$$

$$f(x) = \Omega(g(x)) \Leftrightarrow (\exists \gamma > 0)(\exists x_0 > 0) : x > x_0 \Rightarrow f(x) > \gamma g(x)$$

$$f(x) = \omega(g(x)) \Leftrightarrow (\forall \gamma > 0)(\exists x_0 > 0) : x > x_0 \Rightarrow f(x) > \gamma g(x)$$

They can also be determined using limits[1]:

$$f(x) = O(g(x)) \Leftarrow \lim_{x\to\infty} f(x)/g(x) < \infty$$

$$f(x) = o(g(x)) \Leftrightarrow \lim_{x\to\infty} f(x)/g(x) = 0$$

$$f(x) = \Omega(g(x)) \Leftarrow \lim_{x\to\infty} f(x)/g(x) > 0$$

$$f(x) = \omega(g(x)) \Leftrightarrow \lim_{x\to\infty} f(x)/g(x) = \infty$$

recalling that the definition of limit (for $x \to \infty$) is:

$$\lim_{x\to\infty} f(x) = y \Leftrightarrow (\forall \gamma > 0)(\exists x_0) : x > x_0 \Rightarrow |f(x) - y| < \gamma$$

$$\lim_{x\to\infty} f(x) = \infty \Leftrightarrow (\forall \gamma > 0)(\exists x_0) : x > x_0 \Rightarrow f(x) > \gamma$$

Meanwhile, $\Theta(\cdot)$ is defined as:

$$f(x) = \Theta(g(x)) \Leftrightarrow f(x) = O(g(x)) \ \wedge \ f(x) = \Omega(g(x))$$

[1] A complete definition for $O(\cdot)$ and $\Omega(\cdot)$ would use instead lim sup and lim inf, but this is not necessary for the functions we are interested in.

or

$$f(x) = \Theta(g(x)) \Leftrightarrow (\exists \gamma, \tilde{\gamma} > 0)(\exists x_0 > 0) : x > x_0 \Rightarrow \gamma g(x) < f(x) < \tilde{\gamma} g(x)$$

or

$$f(x) = \Theta(g(x)) \Leftarrow \lim_{x \to \infty} f(x)/g(x) \in (0, \infty)$$

The definition of $\sim$ is similar to the definition of Θ, but it also determines the constant factor:

$$f(x) \sim g(x) \Leftrightarrow \lim_{x \to \infty} f(x)/g(x) = 1$$

These definitions can also be modified to the case when $x \to 0$ instead of $x \to \infty$. The only change in the definitions is that the limits are now $\lim_{x \to 0}$ and the condition on x is now $x < x_0$. However, the relationship between functions change: While $x = O(x^2)$ when $x \to \infty$, this is not true when $x \to 0$ (in this case the opposite would be true: $x^2 = O(x)$). Analogous modifications can be used to analyze functions of two variables $f(x, y)$.

Appendix C

The A_n constant

A_n is the $(n-1)$-dimensional hyper-area of the border of an n-dimensional hyper-sphere of radius 1. $A_1 = 2$, $A_2 = 2\pi$, $A_3 = 4\pi$, and so on. The n-dimensional hyper-volume is A_n/n. This constant can be computed recursively using $A_{n+2} = 2\pi A_n/n$, since:

$$\begin{aligned}\frac{A_{n+2}}{n+2} &= \iint_{x^2+y^2<1} \frac{A_n}{n}(1-x^2-y^2)^{n/2}dxdy \\ &= \int_0^1 \frac{A_n}{n}(1-r^2)^{n/2} \cdot 2\pi r dr \\ &= \int_0^1 \frac{A_n}{n}(1-u)^{n/2} \cdot \pi du = \frac{A_n}{n}\frac{\pi}{\frac{n}{2}+1} \Rightarrow A_{n+2} = 2\pi A_n/n\end{aligned}$$

Solving the recurrence gives us

$$A_n = \frac{2\pi^{n/2}}{(n/2-1)!} \tag{C.1}$$

for even n and

$$A_n = \frac{2\pi^{(n-1)/2}}{(n/2-1)(n/2-2)... \cdot (3/2) \cdot (1/2)} \tag{C.2}$$

for odd n, or $A_n = 2\pi^{n/2}/\Gamma(n/2)$ in general, where $\Gamma(t)$ is the Gamma function[1].

From Equations C.1 and C.2, one can derive that, for integer k:

$$A_{2k}A_{2k+1} = 2^{2k}A_{4k}$$

$$A_{2k}A_{2k-1} = 2^{2k-1}A_{4k-2}$$

and therefore for integer n:

$$A_nA_{n+1} = 2^nA_{2n} \tag{C.3}$$

[1] $\Gamma(t) \triangleq \int_0^\infty x^{t-1}e^{-x}dx$ is a continuous function satisfying $\Gamma(x) = (x-1)!$ for integer x, $\Gamma(x) = (x-1)\Gamma(x-1)$ in general and $\Gamma(1/2) = \sqrt{\pi}$.

We can also show that:

$$\frac{A_n}{A_{n-1}} \sim \sqrt{\frac{2\pi}{n}} \text{ (as } n \to \infty) \tag{C.4}$$

using that:

$$A_n/n = \int_{-1}^{1} \frac{A_{n-1}}{n-1}(1-r^2)^{\frac{n-1}{2}} dr$$

$$= \int_{-1}^{1} \frac{A_{n-1}}{(n-1)^{3/2}} \left(1 - \frac{((\sqrt{n-1})r)^2}{n-1}\right)^{\frac{n-1}{2}} \sqrt{n-1} dr$$

$$= \int_{-\sqrt{n-1}}^{\sqrt{n-1}} \frac{A_{n-1}}{(n-1)^{3/2}} \left(1 - \frac{u^2}{n-1}\right)^{\frac{n-1}{2}} du$$

$$\sim \int_{-\infty}^{\infty} \frac{A_{n-1}}{(n-1)^{3/2}} e^{-u^2/2} du = \frac{\sqrt{2\pi} A_{n-1}}{(n-1)^{3/2}}$$

$$\Rightarrow A_n/A_{n-1} \sim \sqrt{\frac{2\pi}{n}}$$

Appendix D

Fast Greedy

While costing in general $O(N^2 \log N)$ time and $O(N^2)$ memory, the Greedy #2 method can also be solved in $O(N \log N)$ time and $O(N)$ memory when $n = 1$ and cost is Euclidean distance.

D.1 $O(N^2)$ time, $O(N)$ memory version

Before explaining the $O(N \log N)$ algorithm, let us first present an $O(N^2)$ version.

Let $Q \subset \mathbb{R} \times \{1, 2\}$ be a set containing all the points of P_1 and P_2, with a label describing to which one of the two each point belongs, i.e. $Q = \{(x_1, 1) : x_1 \in P_1\} \cup \{(x_2, 2) : x_2 \in P_2\}$.

The $O(N^2)$ algorithm builds this set Q first and sorts it according to the real component (in $O(N \log N)$ time). The optimization it does in relation to the original greedy algorithm is to observe that, in each iteration, the closest pair of points (x_1, x_2) appears necessarily consecutively in Q. So in each iteration, the algorithm traverses the sorted set Q and removes the closest pair of consecutive points of different labels (i.e. one originally belonging to P_1 and the other to P_2) found, adding it to the match set S. Traversal and removal can be implemented in $O(N)$ time using vector or linked list data structures, totaling $O(N^2)$ time.

D.1.1 $O(N \log N)$ time, $O(N)$ memory version

The total cost can be further reduced to $O(N \log N)$ by observing that, between two iterations of the algorithm above, the sorted set Q barely changes, i.e. traversing the whole set Q again wastes a lot of computation time.

This version of the algorithm starts by building Q and sorting it in $O(N \log N)$. Then, it constructs a doubly linked list of it. Also, it constructs a heap data structure, where each node corresponds to a pair of consecutive members of the linked list with different labels (one originally belonging to P_1 and the other to P_2).

The heap compares its nodes according to the distance between the consecutive members, so that the top node is the closest pair of consecutive points with different labels.

The algorithm also requires that each node of the heap have a pointer to its position in the linked list and vice-versa, so that a point can be found in the other data structure in $O(1)$ time. To this end, the heap must be designed in a way that the pointers are updated correctly whenever the linked list is changed, which however does not change the complexity of the operations.

Each iteration of the algorithm will:

- Remove from the heap the top pair (a, b) (in $O(\log N)$ time) and add it to the match set S. Let us use the notation (a, b) to denote consecutive points $a \leq b$, not necessarily satisfying $a \in P_1$ and $b \in P_2$.
- Let a^{left} be the point on the left of a in Q and b^{right} be the point on the right of b in Q. If the pairs (a^{left}, a) and/or (b, b^{right}) are in the heap, remove them from the heap ($O(\log N)$).
- Remove a and b from the linked list ($O(1)$).
- If a^{left} and b^{right} have different labels, add the pair to the heap ($O(\log N)$).

Therefore, each iteration costs $O(\log N)$, which gives a total cost of $O(N \log N)$.

Appendix E

Monte-Carlo solution of "max-prob"

When the log-probabilities cannot be analytically computed to build the cost matrix of the "max-prob" problem, we can efficiently sample them using a Monte-Carlo method.

Recall that the entries of the cost matrix of the "max-prob" problem are:

$$C_{ij} = -\log \mathrm{pdf}[X_1^i, X_2^j | \Pi_{ij} = 1]$$

$$= -\log \int_{\mathbb{R}^n} p(x) g_\epsilon(X_1^i - x) g_\epsilon(X_2^j - x) dx$$

which can be rewritten as:

$$C_{ij} = -\log \int_{\mathbb{R}^n} p(x) g_{\sqrt{2}\epsilon}(X_1^i - X_2^j) g_{\epsilon/\sqrt{2}}\left(x - \frac{X_1^i + X_2^j}{2}\right) dx$$

$$= -\log \left(g_{\sqrt{2}\epsilon}(X_1^i - X_2^j) \int_{\mathbb{R}^n} p(x) g_{\epsilon/\sqrt{2}}\left(x - \frac{X_1^i + X_2^j}{2}\right) dx \right)$$

$$= -\log \left(g_{\sqrt{2}\epsilon}(X_1^i - X_2^j) \int_{\mathbb{R}^n} p\left(\frac{X_1^i + X_2^j}{2} + Z\right) g_{\epsilon/\sqrt{2}}(Z)\, dZ \right)$$

$$\approx -\log \left(g_{\sqrt{2}\epsilon}(X_1^i - X_2^j) \frac{\sum_k p\left(\frac{X_1^i + X_2^j}{2} + Z_k\right)}{\#\text{samples}} \right)$$

$$= \frac{||X_1^i - X_2^j||^2}{4\epsilon^2} - \log\left(\sum_k p\left(\frac{X_1^i + X_2^j}{2} + Z_k\right)\right) + const.$$

where $\{Z_k\}$ are i.i.d. random isotropic Gaussian variables with parameter $\epsilon/\sqrt{2}$.

Curiously, we can verify experimentally that applying the same set $\{Z_k\}$ to every i, j instead of sampling each entry independently (i.e., instead of using a set $\{Z_{i,j,k}\}$)

provides us higher hit rates.

E.1 Experiment

We did a synthetic experiment to test whether using independent or correlated samples is better for the Monte-Carlo method presented here. We generated $N = 10$ pairs of points using the generator set model with isotropic Gaussian distributions in $\mathbb{R}^2$ with a noise ratio of $\epsilon/\sigma = .5$, no outliers. We generated 10^6 pairs of sets P_1, P_2 and ran the exact "max-prob" method (i.e. minimum bipartite matching with squared Euclidean distance), the Monte-Carlo method with independent samples ($\{Z_{i,j,k}\}$), the Monte-Carlo method with correlated samples ($\{Z_k\}$) and Greedy #2. The Monte-Carlo methods used both 10 samples per pair (i.e., $k \in \{1, ..., 10\}$).

The average hit count of the exact "max-prob" method was $E[\#\text{hits}_{\text{exact-MP}}] = 4.6963 \pm 0.0060$ (where, in the notation $A \pm B$, A estimates $E[X]$ and B estimates $3\sqrt{\frac{\text{Var}[X]}{\#\text{samples}}}$). The difference between exact "max-prob" and Monte-Carlo with correlated samples was $E[\#\text{hits}_{\text{exact-MP}} - \#\text{hits}_{\text{MC-correlated}}] = -0.000078 \pm 0.000409$; the difference between exact "max-prob" and Monte-Carlo with independent samples was $E[\#\text{hits}_{\text{exact-MP}} - \#\text{hits}_{\text{MC-independent}}] = -0.0073 \pm 0.0022$; and the difference between exact "max-prob" and Greedy #2 was $E[\#\text{hits}_{\text{exact-MP}} - \#\text{hits}_{\text{Greedy\#2}}] = -0.8449 \pm 0.0058$.

The experiment shows that exact "max-prob" and Monte-Carlo with correlated samples produce higher hit counts than Monte-Carlo with independent samples, which in turn has a higher hit count than Greedy #2. Meanwhile, no significant difference in hit count was found between exact "max-prob" and Monte-Carlo with correlated samples.

Appendix F

Probabilistic Point Querying

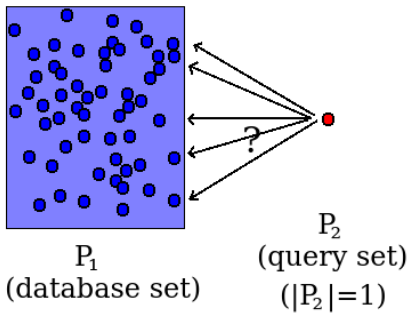


Figure F.1: Illustration of the querying problem.

F.1 The Querying Problem

A related problem to that of "probabilistic point matching" is what we call "*probabilistic point querying*". Perhaps because both problems are typically solved in the Computer Vision literature by assigning a point to its nearest neighbor, both are usually called "matching", although they refer to very different applications.

While in matching we have two sets of N points and we want a 1-to-1 assignment, in querying we may consider that the first set has N points, while the second has only 1 point, and we would like to discover which point from the first set is most likely to correspond to the point of the second set (Figure F.1). This problem arises for instance in recognition applications where a descriptor is queried in a database in order to perform classification.

F.1.1 Probabilistic model

We will use a generator set model where the generator set P is represented by a matrix $\tilde{X} \in \mathbb{R}^{n \times N}$; the first set (database set), P_1, is represented by a matrix

$X \in \mathbb{R}^{n \times N}$, the second set (query set) P_2 has a single point $x' \in \mathbb{R}^n$, and noise is represented by matrices Y_1, Y_2:

$$X = \tilde{X} + Y_1$$

$$x' = (\tilde{X} + Y_2)e_i$$

where $e_i = \begin{bmatrix} 0 & ... & 0 & 1 & 0 & ... & 0 \end{bmatrix}^T$, i.e. $(e_i)_j = 1 \Leftrightarrow i = j$, and 0 otherwise, and i is a uniform random variable in $\{1, ..., N\}$

Also, we assume the columns in $\tilde{X}$, Y_1 and Y_2 are i.i.d. with isotropic Gaussian distributions of parameter σ, ϵ_1 and ϵ_2 respectively[1].

F.1.2 Solution

We can solve this problem by trying to find the most probable i given X and x', i.e.:

$$\arg\max_i P[i|X, x']$$

$$= \arg\max_i \frac{\text{pdf}[X, x'|i]P[i]}{P[X, x']}$$

$$= \arg\max_i \frac{\text{pdf}[x'|i, X]\text{pdf}[X|i]P[i]}{\text{pdf}[X, x']}$$

$$= \arg\max_i \frac{\text{pdf}[x'|i, X]\text{pdf}[X]P[i]}{\text{pdf}[X, x']}$$

$$= \arg\max_i \text{pdf}[x'|i, X]$$

$$= \arg\max_i \text{pdf}[x'|i, X^i]$$

So now we only need to find X^i that maximizes $\text{pdf}[x'|i, X^i]$. In the Gaussian case, this is equal to:

$$\text{pdf}[x'|i, X^i] = \frac{\text{pdf}[x', X^i|i]}{\text{pdf}[X^i|i]} = \frac{\int_{\mathbb{R}^n} g_\sigma(x) g_{\epsilon_1}(X^i - x) g_{\epsilon_2}(x' - x) dx}{g_{\sqrt{\sigma^2+\epsilon_1^2}}(X^i)}$$

$$= \left(\frac{\sqrt{\sigma^2+\epsilon^2}}{2\pi\sigma\epsilon_1\epsilon_2}\right)^n \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}\left(\frac{||x||^2}{\sigma^2} + \frac{||X^i - x||^2}{\epsilon_1^2} + \frac{||x' - x||^2}{\epsilon_2^2} - \frac{||X^i||^2}{\sigma^2+\epsilon_1^2}\right)\right) dx$$

$$= \left(\frac{\sqrt{\sigma^2+\epsilon^2}}{2\pi\sigma\epsilon_1\epsilon_2}\right)^n \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}\begin{bmatrix} x \\ X^i \\ x' \end{bmatrix}^T \begin{bmatrix} \frac{I}{\sigma^2} + \frac{I}{\epsilon_1^2} + \frac{I}{\epsilon_2^2} & -\frac{I}{\epsilon_1^2} & -\frac{I}{\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2} & \frac{I}{\epsilon_1^2} - \frac{I}{\sigma^2+\epsilon_1^2} & 0 \\ -\frac{I}{\epsilon_2^2} & 0 & \frac{I}{\epsilon_2^2} \end{bmatrix} \begin{bmatrix} x \\ X^i \\ x' \end{bmatrix}\right) dx$$

[1]It is expected that the first set (database set) has less noise than the second set (the query set), i.e. $\epsilon_1 \leq \epsilon_2$.

$$= \left(\frac{\sqrt{\sigma^2+\epsilon^2}}{\sqrt{2\pi}\sqrt{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}}\sigma\epsilon_1\epsilon_2} \right)^n \exp\left(-\frac{1}{2} \begin{bmatrix} X^i \\ x' \end{bmatrix}^T A \begin{bmatrix} X^i \\ x' \end{bmatrix} \right)$$

where

$$A = \begin{bmatrix} \frac{I}{\epsilon_1^2} - \frac{I}{\sigma^2+\epsilon_1^2} - \frac{I/\epsilon_1^4}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} & -\frac{I/\epsilon_1^2\epsilon_2^2}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \\ -\frac{I/\epsilon_1^2\epsilon_2^2}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} & \frac{I}{\epsilon_2^2} - \frac{I/\epsilon_2^4}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \end{bmatrix}$$

$$= \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \left(\frac{I}{\epsilon_1^2} - \frac{I}{\sigma^2+\epsilon_1^2}\right)\left(\frac{I}{\sigma^2}+\frac{I}{\epsilon_1^2}+\frac{I}{\epsilon_2^2}\right) - \frac{I}{\epsilon_1^4} & -\frac{I}{\epsilon_1^2\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2\epsilon_2^2} & \frac{\frac{I}{\sigma^2}+\frac{I}{\epsilon_1^2}+\frac{I}{\epsilon_2^2}}{\epsilon_2^2} - \frac{I}{\epsilon_2^4} \end{bmatrix}$$

$$= \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{I}{\epsilon_1^2}\left(\frac{I}{\sigma^2}+\frac{I}{\epsilon_2^2}\right) - \frac{I}{\sigma^2+\epsilon_1^2}\left(\frac{I}{\sigma^2}+\frac{I}{\epsilon_1^2}+\frac{I}{\epsilon_2^2}\right) & -\frac{I}{\epsilon_1^2\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2\epsilon_2^2} & \frac{I}{\sigma^2\epsilon_2^2}+\frac{I}{\epsilon_1^2\epsilon_2^2} \end{bmatrix}$$

$$= \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{I}{\epsilon_1^2}\left(\frac{I}{\sigma^2}+\frac{I}{\epsilon_2^2}\right) - \frac{I}{\sigma^2+\epsilon_1^2}\left(\frac{(\sigma^2+\epsilon_1^2)I}{\sigma^2\epsilon_1^2}+\frac{I}{\epsilon_2^2}\right) & -\frac{I}{\epsilon_1^2\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2\epsilon_2^2} & \frac{I}{\sigma^2\epsilon_2^2}+\frac{I}{\epsilon_1^2\epsilon_2^2} \end{bmatrix}$$

$$= \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{I}{\epsilon_1^2\epsilon_2^2} - \frac{I}{(\sigma^2+\epsilon_1^2)\epsilon_2^2} & -\frac{I}{\epsilon_1^2\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2\epsilon_2^2} & \frac{I}{\sigma^2\epsilon_2^2}+\frac{I}{\epsilon_1^2\epsilon_2^2} \end{bmatrix}$$

$$= \frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{\sigma^2 I}{(\sigma^2+\epsilon_1^2)\epsilon_1^2\epsilon_2^2} & -\frac{I}{\epsilon_1^2\epsilon_2^2} \\ -\frac{I}{\epsilon_1^2\epsilon_2^2} & \frac{(\sigma^2+\epsilon_1^2)I}{\sigma^2\epsilon_1^2\epsilon_2^2} \end{bmatrix}$$

$$= \frac{1/\epsilon_1^2\epsilon_2^2}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{\sigma^2}{(\sigma^2+\epsilon_1^2)}I & -I \\ -I & \frac{(\sigma^2+\epsilon_1^2)}{\sigma^2}I \end{bmatrix}$$

$$= \frac{\frac{\sigma^2+\epsilon_1^2}{\sigma^2\epsilon_1^2\epsilon_2^2}}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}} \begin{bmatrix} \frac{I}{\left(1+\frac{\epsilon_1^2}{\sigma^2}\right)} \\ -I \end{bmatrix} \begin{bmatrix} \frac{I}{\left(1+\frac{\epsilon_1^2}{\sigma^2}\right)} \\ -I \end{bmatrix}^T$$

Therefore,

$$P[x'|i, X^i] = g_{\sigma\epsilon_1\epsilon_2\sqrt{\left(\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}+\frac{1}{\epsilon_2^2}\right)/(\sigma^2+\epsilon_1^2)}}\left(x' - \frac{X^i}{1+\frac{\epsilon_1^2}{\sigma^2}}\right)$$

$$= g_{\sigma\epsilon_1\epsilon_2\sqrt{\frac{1}{\sigma^2\epsilon_1^2}+\frac{1/\epsilon_2^2}{\sigma^2+\epsilon_1^2}}}\left(x' - \frac{X^i}{1+\frac{\epsilon_1^2}{\sigma^2}}\right)$$

$$= g_{\sqrt{\epsilon_2^2+\frac{\sigma^2\epsilon_1^2}{\sigma^2+\epsilon_1^2}}}\left(x' - \frac{X^i}{1+\frac{\epsilon_1^2}{\sigma^2}}\right)$$

$$= g_{\sqrt{\epsilon_2^2+\frac{1}{\frac{1}{\sigma^2}+\frac{1}{\epsilon_1^2}}}}\left(x' - \frac{X^i}{1+\frac{\epsilon_1^2}{\sigma^2}}\right)$$

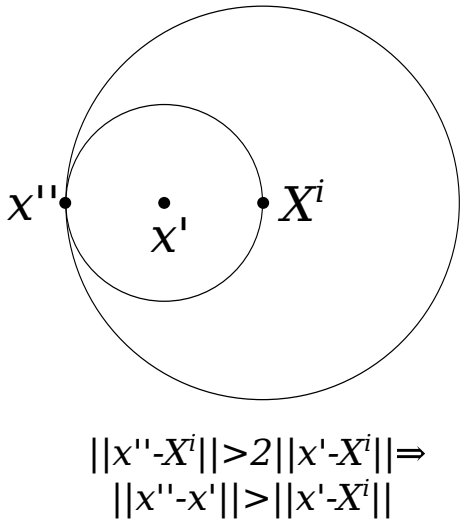


Figure F.2: Illustration of the safety radius $||x'' - X^i|| > 2||x' - X^i||$.

Note from the equation above that $\text{pdf}[x'|i, X^i]$ decreases with $\left\|X^i - \left(1 + \frac{\epsilon_1^2}{\sigma^2}\right) x'\right\|$, and therefore the solution to the querying problem is the nearest point to $\left(1 + \frac{\epsilon_1^2}{\sigma^2}\right) x'$. Note that this is the same as the nearest neighbor solution only if $\epsilon_1 = 0$ (in this case, $\text{pdf}[x'|X^i, i]$ is always a Gaussian distribution centered at X^i, regardless of the distribution of the generator set, and therefore the nearest neighbor solution is also the most probable solution). Curiously, the solution of this model only depends on the noise parameter ϵ_1 of the database set, being indifferent to the noise ϵ_2 on the query set.

F.1.3 Asymptotic behavior

Let us see how the querying problem behaves as N grows to infinity. Let us suppose that $\epsilon_1 = 0$ and that $\epsilon_2 \to 0$ as $N \to \infty$.

The condition for correctly matching x' is that X^i is its nearest neighbor. Therefore the probability of hitting the query is:

$$Q = \iint_{\mathbb{R}^n \times \mathbb{R}^n} P[||x'' - x'|| > ||X^i - x'|| \mid X^i, x']^{N-1} \text{pdf}[X^i, x'] dx' dX^i$$

where x'' is a random point X_j of the database set, with $j \neq i$.

Using the bound illustrated in Figure F.2, we obtain:

$$Q \geq \iint_{\mathbb{R}^n \times \mathbb{R}^n} P[||x'' - X^i|| > 2||X^i - x'|| \mid X^i, x']^{N-1} \text{pdf}[X^i, x'] dx' dX^i$$

We can bound the equation above using the maximum probability $p_0 =$

$\max_x p(x)$ multiplied by the volume of the sphere containing $\tilde{x}''$, obtaining:

$$\begin{aligned} Q &\geq \iint_{\mathbb{R}^n \times \mathbb{R}^n} \max\left\{0, 1 - \frac{p_0 A_n}{n}(2||X^i - x'||)^n\right\}^{N-1} \text{pdf}[x', X^i] dx' dX^i \\ &= E\left[\max\left\{0, 1 - \frac{p_0 A_n (2\epsilon_2)^n}{n}||G||^n\right\}^{N-1}\right] \end{aligned} \tag{F.1}$$

where G is a random isotropic Gaussian variable in $\mathbb{R}^n$ with unitary variance.

We can further bound this to

$$\begin{aligned} Q &\geq E\left[1 - (N-1)\frac{p_0 A_n (2\epsilon_2)^n}{n}||G||^n\right] \\ &= 1 - (N-1)\frac{p_0 A_n (2\epsilon_2)^n}{n} E\left[||G||^n\right] \\ &\geq \bar{Q} \Leftarrow \epsilon_2^n \leq \frac{1 - \bar{Q}}{(N-1)\frac{p_0 A_n 2^n}{n} E\left[||G||^n\right]} \end{aligned}$$

where (Using Equations 7.4 and C.3):

$$E\left[||G||^n\right] = \int_0^\infty A_n \frac{e^{-r^2/2} r^n r^{n-1} dr}{(2\pi)^{n/2}} = \frac{A_n}{(2\pi)^{n/2}} \frac{(2\pi)^n}{A_{2n}} = \frac{(8\pi)^{n/2}}{A_{n+1}}.$$

Note that analogously to the matching problem, in the querying problem, we can guarantee a minimum probability of correctly matching the query if ϵ_2 satisfies a constraint of the form $\epsilon_2^n \leq C/N$, with $C > 0$. Furthermore, if $\epsilon_2^n = o(1/N)$, the query is correct with 100% probability as $N \to \infty$.

Conversely, when $\epsilon_2^n \sim C/N$, Equation F.1 becomes:

$$Q \geq E\left[\left(1 - \frac{p_0 A_n (2\epsilon_2)^n}{n}||G||^n\right)^{N-1} \middle| \frac{p_0 A_n (2\epsilon_2)^n}{n}||G||^n < 1\right] P\left[\frac{p_0 A_n (2\epsilon_2)^n}{n}||G||^n < 1\right]$$

$$\sim E\left[\left(1 - \frac{p_0 A_n 2^n C}{nN}||G||^n\right)^{N-1} \middle| \frac{p_0 A_n 2^n C}{nN}||G||^n < 1\right] P\left[\frac{p_0 A_n 2^n C}{nN}||G||^n < 1\right]$$

$$\sim E\left[\exp\left(-\frac{p_0 A_n 2^n C}{n}||G||^n\right)\right] \in (0, 1)$$

which means that for every C, there exists $\bar{Q} > 0$ such that $\epsilon_2^n \sim C/N \Rightarrow Q \gtrsim \bar{Q}$.

F.2 The querying problem with outliers

Let us suppose now that there is a probability of q that x' does not match any point in the database set, i.e., that x' is an outlier.

In this case, the generator model is:

$$X = \tilde{X} + Y_1$$

$$x' = (s\tilde{X} + (1-s)\tilde{X}' + Y_2)e_i$$

where $\tilde{X}'$ has the same distribution of $\tilde{X}$ and s is a Bernoulli random variable with $P[s=0] = q$ and $P[s=1] = 1-q$.

F.2.1 Solution

We can solve this problem in two steps: First determine if x' is an inlier or not, and afterwards determine which point X^i in the database set corresponds to it.

So first we compute:

$$\begin{aligned}
P[s=1|X,x'] &= \frac{P[X,x'|s=1]P[s=1]}{P[X,x'|s=1]P[s=1] + P[X,x'|s=0]P[s=0]} \\
&= 1/\left(1 + \frac{P[X,x'|s=0]P[s=0]}{P[X,x'|s=1]P[s=1]}\right) \\
&= 1/\left(1 + \frac{P[s=0]/P[s=1]}{P[X,x'|s=1]/P[X,x'|s=0]}\right) \\
&= 1/\left(1 + \frac{q/(1-q)}{P[X,x'|s=1]/(P[X]P[x'])}\right) \\
&= 1/\left(1 + \frac{q/(1-q)}{P[x'|X,s=1]/P[x']}\right) \\
&= 1/\left(1 + \frac{q/(1-q)}{\left(\sum_i P[x'|s=1,X,i]P[i]\right)/P[x']}\right) \\
&= 1/\left(1 + \frac{q/(1-q)}{\left(\sum_i \frac{1}{N}\frac{P[x'|s=1,X^i,i]}{P[x']}\right)}\right)
\end{aligned}$$

Therefore, we can choose to determine that x' is an inlier if $P[s=1] > .5$, which happens if and only if:

$$\frac{1}{N}\sum_i \frac{P[x'|s=1,X^i,i]}{P[x']} > \frac{q}{1-q}$$

$$\Leftrightarrow \frac{1}{N}\sum_i \zeta\left(X^i, x'\right) > \frac{q}{1-q}$$

where $\zeta(\cdot,\cdot)$ is analogous to the normalized joint probability #2 defined in Section 4.1.3; in this case $\zeta(X^i,x') = \frac{\text{pdf}[X^i,x'|i,s=1]}{\text{pdf}[X^i]\text{pdf}[x']}$.

If we found that probably $s=1$, then the next step would be to determine i.

Given that $s = 1$, this step is identical to the case without outliers. Curiously, given that $s = 0$, all values of i have equal probability, and therefore

$$\arg\max_i P[i|x', X] =$$

$$\arg\max_i P[i|x', X, s = 1]P[s = 1] + P[i|x', X, s = 0]P[s = 0]$$

$$= \arg\max_i P[i|x', X, s = 1]$$

F.3 Experiments

F.3.1 Comparison with nearest neighbor

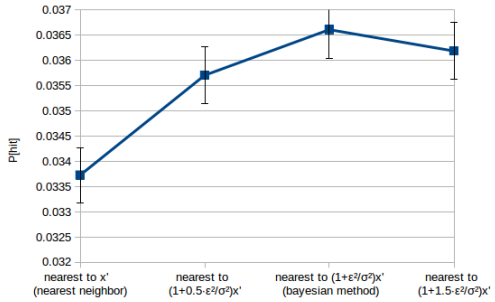
The purpose of this experiment is to show that the Bayesian method derived in Section F.1.2 is superior to the nearest neighbor method when $\epsilon_1 \neq 0$. We generated 10^6 pairs of sets P_1, P_2 with $N = 100$ points, a noise ratio of $\epsilon_1/\sigma = \epsilon_2/\sigma = .5$, no outliers, and computed the hit ratio for different querying criteria.

The different querying criteria are: closest point to x' (nearest neighbor), closest to $(1 + 0.5\epsilon^2/\sigma^2)x'$, closest to $(1 + \epsilon^2/\sigma^2)x'$ (Bayesian method), and closest to $(1 + 1.5\epsilon^2/\sigma^2)x'$. Figure F.3 compares the hit rate for $n \in \{1, 2, 4, 10\}$. The error bars display an error of the form $\pm 3\sqrt{\frac{\text{Var}[X]}{\#\text{samples}}}$. The figure suggests that the Bayesian method has the highest hit rates, particularly in high-dimensional cases.

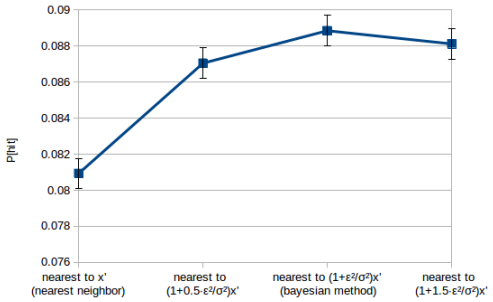
F.3.2 Asymptotic behavior

In this experiment we evaluate the condition for constant hit rate derived in this chapter. We ran the nearest neighbor and Bayesian method on sets generated using Gaussian distributions with $\epsilon_1/\sigma = \epsilon_2/\sigma = .5/N^{1/n}$ (Figure F.4) and $\epsilon_1 = 0$, $\epsilon_2/\sigma = .5/N^{1/n}$ (Figure F.5), for varying N and n. The Figures suggest that $\epsilon^n = O(1/N)$ yields a minimum expected hit rate, in agreement with the derived condition. Additionally, we observe that the hit rate is very similar for both methods, being slightly higher for the Bayesian method when $\epsilon_1 = \epsilon_2$ and higher for the nearest neighbor method[2] when $\epsilon_1 = 0$.

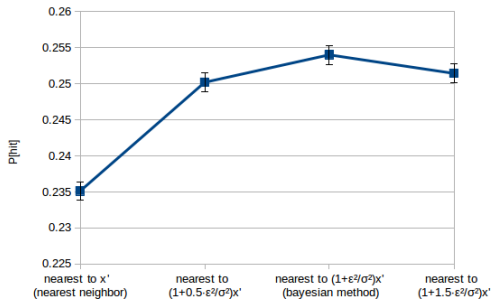
[2]When $\epsilon_1 = 0$, the Bayesian method is equal to the nearest neighbor method. Choosing the closest point to $(1 + \epsilon_2^2/\sigma^2)x'$ is not the Bayesian method.



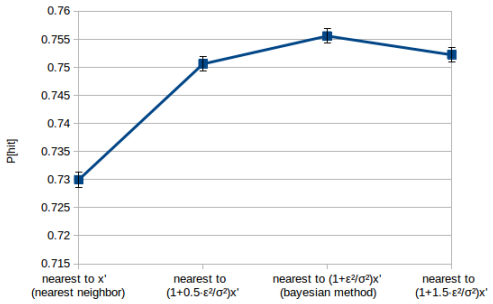
(a) when $n = 1$



(b) when $n = 2$

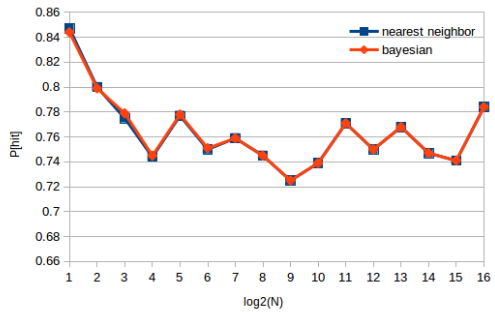


(c) when $n = 4$

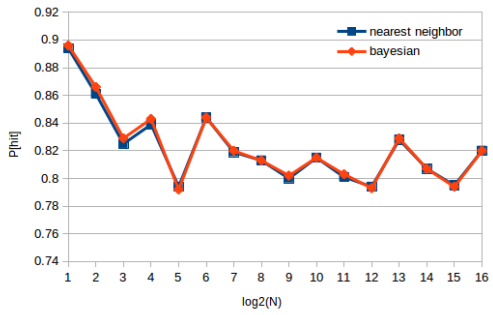


(d) when $n = 10$

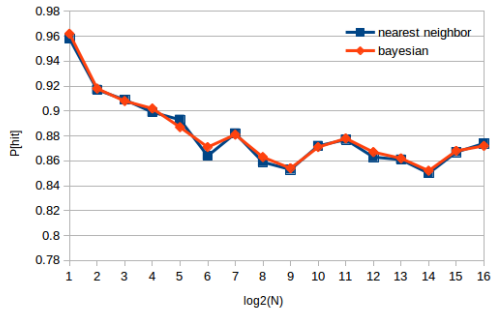
Figure F.3: Hit rate of different querying criteria.



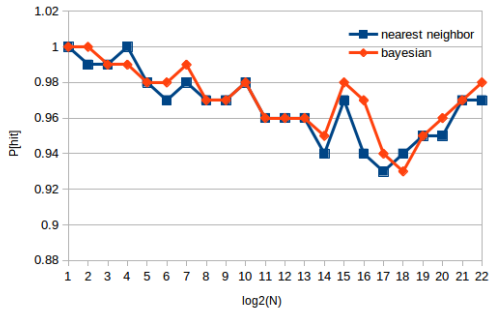
(a) when $n = 1$, 1000 samples per case



(b) when $n = 2$, 1000 samples per case

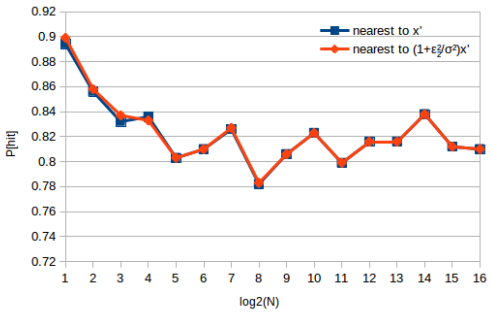


(c) when $n = 4$, 1000 samples per case

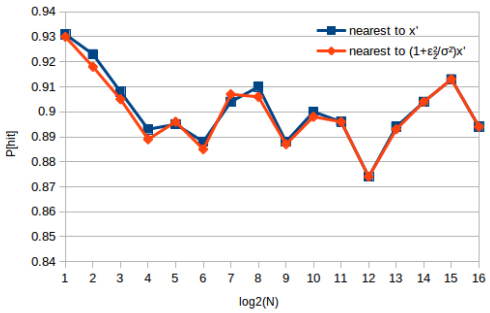


(d) when $n = 10$, 100 samples per case

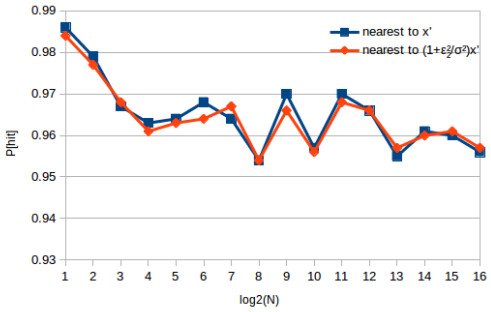
Figure F.4: Hit rate when $\epsilon_1^n = \epsilon_2^n = \Theta(1/N)$.



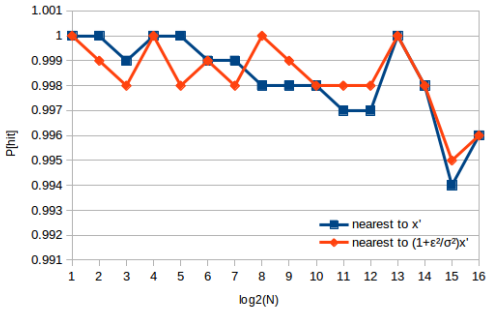
(a) when $n = 1$, 1000 samples per case



(b) when $n = 2$, 1000 samples per case



(c) when $n = 4$, 1000 samples per case



(d) when $n = 10$, 1000 samples per case

Figure F.5: Hit rate when $\epsilon_1 = 0$ and $\epsilon_2^n = \Theta(1/N)$.

Bibliography

[1] SZELISKI, R. *Computer Vision: Algorithms and Applications*. 1st ed. New York, NY, USA, Springer-Verlag New York, Inc., 2010. ISBN: 1848829345, 9781848829343.

[2] FELZENSZWALB, P. F., HUTTENLOCHER, D. P. "Efficient belief propagation for early vision", *International journal of computer vision*, v. 70, n. 1, pp. 41–54, 2006.

[3] TOLA, E., LEPETIT, V., FUA, P. "Daisy: An efficient dense descriptor applied to wide-baseline stereo", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 32, n. 5, pp. 815–830, 2010.

[4] BESL, P. J., MCKAY, N. D. "Method for registration of 3-D shapes". In: *Robotics-DL tentative*, pp. 586–606. International Society for Optics and Photonics, 1992.

[5] RUSU, R., BLODOW, N., MARTON, Z., et al. "Aligning point cloud views using persistent feature histograms". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3384–3391, Sept 2008. doi: 10.1109/IROS.2008.4650967.

[6] LI, X., GUSKOV, I. "Multi-scale Features for Approximate Alignment of Point-based Surfaces". In: *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. ISBN: 3-905673-24-X.

[7] VEENMAN, C. J., REINDERS, M., BACKER, E. "Resolving Motion Correspondence for Densely Moving Points", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, pp. 54–72, 2001.

[8] BERCLAZ, J., FLEURET, F., TURETKEN, E., et al. "Multiple object tracking using k-shortest paths optimization", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 33, n. 9, pp. 1806–1819, 2011.

[9] SHAFIQUE, K., SHAH, M. "A noniterative greedy algorithm for multiframe point correspondence", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 27, n. 1, pp. 51–65, 2005.

[10] BELONGIE, S., MALIK, J., PUZICHA, J. "Shape matching and object recognition using shape contexts", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 24, n. 4, pp. 509–522, 2002.

[11] JAIN, A. K., PRABHAKAR, S., HONG, L., et al. "Filterbank-based fingerprint matching", *Image Processing, IEEE Transactions on*, v. 9, n. 5, pp. 846–859, 2000.

[12] TICO, M., KUOSMANEN, P. "Fingerprint matching using an orientation-based minutia descriptor", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 25, n. 8, pp. 1009–1014, 2003.

[13] MÉZARD, M., PARISI, G. "The Euclidean matching problem", *Journal de Physique*, v. 49, n. 12, pp. 2019–2025, 1988.

[14] SICURO, G., CARACCIOLO, S. *The Euclidean Matching Problem*. Tese de Doutorado, Università di Pisa, Pisa, Italy, 2014.

[15] LOWE, D. G. "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, v. 60, n. 2, pp. 91–110, 2004.

[16] MIKOLAJCZYK, K., SCHMID, C. "A performance evaluation of local descriptors", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 27, n. 10, pp. 1615–1630, 2005.

[17] MUJA, M., LOWE, D. G. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration." .

[18] BROWN, M., SZELISKI, R., WINDER, S. "Multi-image matching using multi-scale oriented patches". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, v. 1, pp. 510–517. IEEE, 2005.

[19] BURKARD, R. E., DELL'AMICO, M., MARTELLO, S. *Assignment Problems, Revised Reprint*. Siam, 2009.

[20] GOLDBARG, M. C., LUNA, H. P. L. *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. 1st ed. Rio de Janeiro, RJ, Brazil, Editora Campus, 2000.

[21] THULASIRAMAN, K., SWAMY, M. N. *Graphs: theory and algorithms*. John Wiley & Sons, 2011.

[22] CONTE, D., FOGGIA, P., SANSONE, C., et al. "Thirty Years of Graph Matching in Pattern Recognition". 2004.

[23] FISCHLER, M. A., BOLLES, R. C. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, v. 24, n. 6, pp. 381–395, 1981.

[24] JERRUM, M., SINCLAIR, A., VIGODA, E. "A Polynomial-time Approximation Algorithm for the Permanent of a Matrix with Non-negative Entries". In: *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pp. 712–721, New York, NY, USA, 2001. ACM. ISBN: 1-58113-349-9. doi: 10.1145/380752.380877.

[25] GLYNN, D. G. "The permanent of a square matrix", *European Journal of Combinatorics*, v. 31, n. 7, pp. 1887 – 1891, 2010. ISSN: 0195-6698. doi: http://dx.doi.org/10.1016/j.ejc.2010.01.010.

[26] VILLANI, C. *Optimal transport: old and new*, v. 338. Springer Science & Business Media, 2008.

[27] HARRIS, C., STEPHENS, M. "A combined corner and edge detector." In: *Alvey vision conference*, v. 15, p. 50. Citeseer, 1988.

[28] ARANDJELOVIĆ, R., ZISSERMAN, A. "Three things everyone should know to improve object retrieval". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2911–2918. IEEE, 2012.

[29] KE, Y., SUKTHANKAR, R. "PCA-SIFT: A more distinctive representation for local image descriptors". In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, v. 2, pp. II–506. IEEE.