DELAY-BASED METHODS FOR
ROBUST GEOLOCATION OF INTERNET HOSTS

by

Inja Youn
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

| | |
|---|---|
| _____ | Dr. Dana Richards, Dissertation Director |
| _____ | Dr. Brian L. Mark, Dissertation Co-Director |
| _____ | Dr. Kris Gaj, Committee Member |
| _____ | Dr. Daniel B. Carr, Committee Member |
| _____ | Dr. Sanjeev Setia, Department Chair |
| _____ | Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering |

Date: _____ Spring Semester 2013
George Mason University
Fairfax, VA

Delay-Based Methods for Robust Geolocation of Internet Hosts

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Inja Youn
Master of Science
George Mason University, 2004

Dissertation Director: Dr. Dana Richards, Associate Professor
Department of Computer Science
Dissertation Co-Director: Dr. Brian L. Mark, Professor
Department of Electrical and Computer Engineering

Spring Semester 2013
George Mason University
Fairfax, VA

# Table of Contents

# List of Tables

# List of Figures

# Abstract

DELAY-BASED METHODS FOR ROBUST GEOLOCATION OF INTERNET HOSTS

Inja Youn, PhD

George Mason University, 2013

Dissertation Directors: Dr. Dana Richards and Dr. Brian L. Mark

In the past few years, there has been a growing need for accurate geolocation of IP addresses, which is now a must-have feature of many Internet applications. Automated geolocation of IP addresses has important applications, including targeted delivery of localized content over Internet (news, weather, advertising, restriction of localized content based on regional policies, etc.), prevention of Internet crimes (credit card and bank fraud, identity theft, spam, phishing, etc.), detection and prevention of cyberattacks and cyberterrorism, etc. The current geolocation algorithms can be divided into several classes according to the data that is used for determining the geographic location: database-based (which use a database of mappings between Internet prefixes and their corresponding geographical locations), pure-delay based (which take as input is the round trip delay of the probing hosts which are called landmarks), location-delay based (which use the information about both the geographical location and the probing hosts), supplementary information based (which in addition to delay and geographical location, use other available information, such as DNS parsing, geographical and demographical data, etc.).

However, use of network delay time for geolocation has proved not very reliable in the past, because of the non-linear correlation between distances and delays generated by the network congestion, queuing delay and circuitous routes. This thesis brings important advancements to two classes of geolocation methods. The first advancement is a family of pure delay-based algorithms based on a general class of proximity measures. When such measures are carefully chosen to discard the data which contains little information about the geographical location of a target IP address, the resulting algorithms have improved accuracy over the existing pure-delay based schemes. The second advancement, belonging to the location-delay based class of algorithms, is the development of a statistical geolocation scheme based on the application of kernel density estimation to delay measurements amongst a set of landmarks. An estimate of the target IP location is then obtained by maximizing the likelihood of the distances from the target to the landmarks, given the measured delays. This is achieved by an algorithm which combines gradient ascent and force-directed methods. We compare the proposed geolocation schemes with the previous methods by developing a measurement framework based on PlanetLab infrastructure and we compare the experimental geolocation error for the proposed algorithms compared with that for the existing schemes. We find the proposed geolocation algorithms have superior accuracy to the previously developed ones.

# Chapter 1: Introduction

In the past years, there is an increasing number of location-aware application for both mobile and fixed IP addresses. These applications use the location information for collecting user data and also for providing location-based services. Thus, geolocation of IP addresses has widespread important uses, which include:

1. Targeted delivery of local news, weather, advertisement and other content. Thus, for a visitor from a specific city, a web service can deliver news, embed advertisements, and weather forecasts for the specific location

2. Restricting digital content and sales to authorized regions, in conformity with company policies and local law

3. Cloud computing, where some of the organizations have to ensure that their content stays in the appropriate geographic region

4. Determining the regional distribution of clients (by analyzing the Web logs), which offers important marketing information

5. Prevention and reduction of Internet frauds, such as credit card fraud, identity theft, spam and phishing. Thus, when a service request comes from a suspect location, it can be filtered and thoroughly analyzed. Thus, a service provider (e.g. bank) usually constructs a profile of each user, based on the geographical locations from where the user accessed the service. When a user presents the credentials to the service provider from a different location, the service provider can take preemptive measures to protect against the potential phishing attacks, such as temporary blocking client's account, verification of user's location by an alternative communication method (e.g. phone), and, if necessary, requiring the user to change his/her account credentials.

6. Applications in intrusion detection and prevention of cyberterrorism. Detecting and eliminating (or at least mitigating) the Internet attacks are a high-priority national security goal, and IP geolocation is an important instrument for visualizing the overall situation and identifying the attackers.

The traditional approach to IP geolocation is to construct and maintain large databases between IP subnets and their geographical location. For this approach, the geolocation process consists of a simple lookup of a subnet in the database, with the return of the estimate location. Examples of such databases which provide mapping and lookup services include: Quova, IP2Location, MaxMind, IPLigence, GeoBytes, and NetAcuity.

However, this database approach has a few obvious disadvantages. First, due to the dynamic structure of IP addressing, these databases need to be updated on a continuous basis. The algorithms used to update these databases are proprietary and of questionable accuracy, as outlined in [1]. Second, this subnet-mapping approach becomes prohibitive to update and maintain with the advent and adoption of IPv6. Moreover, these databases cannot adapt easily to the frequent location changes of mobile targets. Thus, there is a need to have an algorithm which can estimate the location of an IP target on-the-fly by using delay measurements from a set of hosts, their geographical location, and eventually supplementary information (DNS hints, population distribution, etc.).

There have been numerous efforts to automate the geolocation of IP addresses. However, using the distance and delays for accurate geolocation has been proved an elusive goal, due to the nonlinear correlation between the distance and network delay generate by differences in network latency and circuitous paths. This leads to unacceptable geolocation errors of sometimes more than 1000 km of the previous algorithms.

In this thesis, we bring a two-fold contribution in the field of automated IP geolocation. First, we propose a general framework for specifying a class of delay-based IP geolocation algorithms, which includes GeoPing and Shortest Ping as special cases. The proposed framework is based on a proximity measure consisting of a distance metric together with a norm. The proximity measure is applied to a set of delay measurements obtained from

a set of geographically distributed landmark nodes, and used to determine the landmark in closest proximity to the target. We consider various combinations of distance metrics and norms to obtain proximity measures which result in different geolocation performance characteristics. In particular, we are interested in proximity measures that are robust to measurement errors and reduce the negative influence of landmarks with large delay measurements to the target. We present an extensive set of experimental results to evaluate and compare the performance of delay-based geolocation algorithms derived from different proximity measures, including the Shortest Ping and GeoPing methods as special cases. A major outcome of our empirical study is a recommendation on the proximity measures that yield the highest levels of geolocation accuracy and robustness.

Second, we develop a statistical geolocation scheme based on applying kernel density estimation to delay measurements obtained among a set of landmarks. An estimate of the target location is then obtained by maximizing the likelihood of the distances from the target to the landmarks, given the measured delays. This is achieved by an algorithm which combines gradient ascent and force-directed methods. We present experimental results to demonstrate the superior accuracy of the proposed geolocation scheme compared to previous methods.

# Chapter 2: Background

In this chapter, we discuss the background and related work on measurement-based IP geolocation algorithms. TODO: make three sentences here.

## 2.1 Pure delay-based geolocation schemes

Shortest Ping and GeoPing [2] are two of the earliest IP geolocation algorithms based on delay measurements. Both schemes involve a set of hosts with known locations called *landmarks*, which obtain delay measurements by transmitting ICMP ping packets to the target and possibly other landmarks. The delay measurements are then used to determine the landmark that is in closest proximity to the target. The location of target is then approximated by that of the selected landmark.

More precisely, let $\mathcal{L}_a$ denote an index set for the landmarks, so that the set of landmarks is given by $\{L_i : i \in \mathcal{L}_a\}$. The total number of landmarks is denoted by $|\mathcal{L}_a|$, where in this context $|\cdot|$ denotes set cardinality. The latitude and longitude of landmark $L_i$ in units of radians are denoted by $\phi_i$ and $\lambda_i$, respectively. Let $d_{i\tau}$ denote the round-trip time (RTT) delay measured between landmark $L_i$ and the target $\tau$. In the Shortest Ping scheme, the location estimate for the target is given by $(\phi_k, \lambda_k)$, where

$$k = \arg \min_{i \in \mathcal{L}_a} d_{i\tau}. \tag{2.1}$$

The delay value $d_{i\tau}$ can be interpreted as a measure of proximity of the target to landmark $L_i$. Thus, in Shortest Ping, the landmark in closest proximity to the target is defined to be the one corresponding to the minimum measured delay to the target.

The GeoPing scheme involves a set of nodes called *passive* landmarks, in addition to

the *active* landmarks employed in Shortest Ping. Like the active landmarks, the passive landmarks have known locations, but the passive landmarks do not perform delay measurements. On the other hand, the active landmarks perform delay measurements between each other, as well as to the passive landmarks. Let $\mathcal{L}_a$ denote the index set for active landmarks and let $\mathcal{L}_p$ denote the index set for passive landmarks. The number of active and passive landmarks are denoted by $|\mathcal{L}_a|$ and $|\mathcal{L}_p|$, respectively. The index set for all landmarks is given by $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$.

Let $d_{ij}$ denote the measured delay between landmarks $L_i$ and $L_j$, where $i \in \mathcal{L}_a$ and $j \in \mathcal{L}$. In GeoPing, the location estimate for the target $\tau$ is defined as $(\phi_k, \lambda_k)$, where

$$k = \arg \min_{j \in \mathcal{L}} \sum_{i \in \mathcal{L}_a} (d_{ij} - d_{i\tau})^2. \tag{2.2}$$

Here, the measure of proximity of the target to landmark $L_j$ is determined by the value of $\sum_{i \in \mathcal{L}_a} (d_{ij} - d_{i\tau})^2$, which can be interpreted as the squared Euclidean distance between the vectors $(d_{ij} : i \in \mathcal{L}_a)$ and $(d_{i\tau} : i \in \mathcal{L}_a)$ in $\mathbb{R}^{|\mathcal{L}_a|}$. Intuitively speaking, the location estimate in GeoPing is taken to be the location of the landmark whose delay measurements to the set of active landmarks, $\mathcal{L}_a$, are most "similar" to the corresponding delay measurements between the active landmarks and the target $\tau$, as defined by the proximity values.

Various subsequent studies [3–8] have shown that GeoPing can suffer from poor accuracy. The GeoPing algorithm does not penalize landmarks that are far away and do not carry information about target location. Furthermore, as we shall see in Section 3.7 of this thesis, the Euclidean distance is not a robust metric for geolocation. Finally, the original GeoPing scheme does not exploit information that can be obtained from repeated delay measurements. Ziviani *et al.* [3] proposed replacing the minimum delay distance metric and Euclidean norm used in GeoPing with several alternative metrics: the Manhattan (taxicab) norm, Chebyshev norm, Canberra distance, and cosine or correlation distance. In the present work, we shall consider a larger class of proximity measures and incorporate additional information derived from delay measurements to improve the performance of

pure delay-based geolocation.

## 2.2 Incorporating prior information

Various types of prior information can be incorporated into measurement-based IP geolocation algorithms. For example, the geographical location and inter-landmark distances can be used to perform a form of multilateration. Multilateration [9, 10] is the estimation of location of a target using measurements from two or more stations at known location. Gueye *et al.* in their Constraint-Based Geolocation (CBG) algorithm [5] propose using multilateration together with geographic distance constraints to determine the probable location of the target. In [7], we proposed a statistical approach to process the end-to-end delay measurements. The algorithm uses nonparametric kernel density estimation to process the end-to-end measured delays and inter-landmark distances into "landmark profiles." A force-directed algorithm is used to perform a majority maximization of individual likelihoods. Arif *et al.* [11] proposed a variation of this algorithm by replacing the nonparametric density estimation with parametric (log-normal) density estimation, and performing maximum likelihood estimation of the target location.

Besides inter-landmark distance, other information used by existing algorithms include the following:

- Delay to intermediate hops, used by Katz-Basset *et al.* in their Topology-Based Geolocation (TBG) algorithm [6], which attempts to geolocate the intermediate hosts and the target simultaneously.

- DNS location information obtained from parsing the DNS or WHOIS database for geographical information, which can be used either as a standalone method, as in the GeoTrack method of Padmanabhan and Subramanian [2], or for hints and validation of other methods, as in Katz-Basset *et al.* [6].

- The assumption that the hosts from the same subnet are likely co-located. By estimating the true location of a host, we can extrapolate the information to the other

hosts in the same subnet. This is the approach of the GeoTrack and GeoCluster algorithms of Padmanabhan and Subramanian [2].

- Non-uniform demographic distribution of the population. This is used by Wong *et al.* in their Octant algorithm [12], which clips out large bodies of water and sparsely-populated regions in order to improve geolocation accuracy.

In the remainder of this work, we shall focus exclusively on two classes of geolocation algorithms. In Chapter 3 we will develop a scheme that generalized the pure delay-based algorithms, such as Shortest Ping and GeoPing schemes. Our objective is to improve the geolocation accuracy of this class of algorithms and thereby close the performance gap relative to the algorithms that make use of supplementary prior information. In Chapter 4 we will formulate for the first time a statistical geolocation algorithm, which replaces the upper bounds constrains used by Constraint-Based Geolocation (CBG) [4, 5] with a likelihood maximization approach. Finally, we present in Chapter 5 a summary of our results and improvements to the previous geolocation schemes.

# Chapter 3: Delay-Based Proximity Measures for Robust IP Geolocation

In this section, we develop a framework that generalizes the proximity measures used in delay-based IP geolocation schemes such as Shortest Ping and GeoPing. TODO: three sentences here

## 3.1 Definition of proximity measure

We assume that a set of $m$ delay measurements is performed between each active landmark $L_i$, $i \in \mathcal{L}_a$, and each landmark (active or passive) $L_j$, $j \in \mathcal{L}$. We define $n = |\mathcal{L}_a|$ and $N = |\mathcal{L}|$. Let $d_{ij}^{(l)}$ represent the $l$th smallest delay value between $L_i$ and $L_j$. Define the vector $\boldsymbol{d}_{ij} = (d_{ij}^{(1)}, d_{ij}^{(2)}, \ldots, d_{ij}^{(m)}) \in \mathbb{R}_+^m$, which represents the set of $m$ delay measurements between $L_i$ and $L_j$, in increasing order, i.e., $d_{ij}^{(1)} \leq d_{ij}^{(2)} \leq \cdots \leq d_{ij}^{(m)}$. Similarly, a set of $m$ delay measurements is performed between each active landmark $L_i$, $i \in \mathcal{L}_a$, and the target $\tau$. Let $d_{i\tau}^{(l)}$ denote the $l$th smallest delay measurement value between $L_i$ and the target, and define the sorted vector $\boldsymbol{d}_{i\tau} = (d_{i\tau}^{(1)}, d_{i\tau}^{(2)}, \ldots, d_{i\tau}^{(m)}) \in \mathbb{R}_+^m$.

Let $\mu$ be a distance metric on $\mathbb{R}^m$, i.e., a real-valued function $\mu(\boldsymbol{x}, \boldsymbol{y})$, defined for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ satisfying the following four properties:

1. $\mu(\boldsymbol{x}, \boldsymbol{y}) \geq 0$;
2. $\mu(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$;
3. Symmetry: $\mu(\boldsymbol{x}, \boldsymbol{y}) = \mu(\boldsymbol{y}, \boldsymbol{x})$;
4. Triangle inequality: $\mu(\boldsymbol{x}, \boldsymbol{y}) \leq \mu(\boldsymbol{x}, \boldsymbol{z}) + \mu(\boldsymbol{z}, \boldsymbol{y})$ , $\boldsymbol{z} \in \mathbb{R}^m$.

In practice, we will often ignore some of the measurements, thus we will usually relax the distance metric requirement, resulting in a *premetric*, which satisfies the following two properties:

1. $\mu(\boldsymbol{x}, \boldsymbol{y}) \geq 0$;
2. $\mu(\boldsymbol{x}, \boldsymbol{x}) = 0$.

For each $j \in \mathcal{L}$, let $\nu_j$ be a norm on $\mathbb{R}^n$, i.e., a function $\nu_j(\boldsymbol{x})$, defined for all $\boldsymbol{x} \in \mathbb{R}^n$, satisfying the following three properties:

1. $\nu_j(a\boldsymbol{x}) = |a|\nu_j(\boldsymbol{x})$ for all $a \in \mathbb{R}$;
2. $\nu_j(\boldsymbol{x} + \boldsymbol{y}) \leq \nu_j(\boldsymbol{x}) + \nu_j(\boldsymbol{y})$;
3. $\nu_j(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = \mathbf{0}$, where $\mathbf{0}$ denotes the zero vector.

As we have done for the distance metric $\mu$, in order to encompass all cases, we relax the requirement of the norm to those of a *seminorm* by replacing 3) with 3') as follows:

3') $\nu_j(\mathbf{0}) = 0$, where $\mathbf{0}$ denotes the zero vector.

Then the proximity of landmark $L_j$, $j \in \mathcal{L}$, to the target $\tau$ is defined by

$$\rho_{j\tau} = \nu_j(\mu(\boldsymbol{d}_{1j}, \boldsymbol{d}_{1\tau}), \mu(\boldsymbol{d}_{2j}, \boldsymbol{d}_{2\tau}), \ldots, \mu(\boldsymbol{d}_{nj}, \boldsymbol{d}_{n\tau})). \tag{3.1}$$

In most of our measures (except for Shortest Ping), we have $\nu_1 = \nu_2 = \ldots = \nu_N \triangleq \nu$, in which case the proximity becomes

$$\rho_{j\tau} = \nu(\mu(\boldsymbol{d}_{1j}, \boldsymbol{d}_{1\tau}), \mu(\boldsymbol{d}_{2j}, \boldsymbol{d}_{2\tau}), \ldots, \mu(\boldsymbol{d}_{nj}, \boldsymbol{d}_{n\tau})). \tag{3.2}$$

The estimate of the target location is then given by the location of landmark $L_k$, where

$$k = \arg\min_{j \in \mathcal{L}} \rho_{j\tau}. \tag{3.3}$$

The GNNDS steps are detailed in Algorithm 1.

**Algorithm 1** GNNDS: Generalized Nearest-Neighbor Delay-Space Algorithm

**G1.** Initialize the minimum proximity landmark index and minimum proximity value:
$\rho_{min} \leftarrow \infty$
$k \leftarrow 0$

**G2.** Calculate the distance between delay sets $d_{ij}$ and $d_{i\tau}$
**for** $i \in \mathcal{L}_a$ **do**
**for** $j \in \mathcal{L}$ **do**
$\delta_{ij\tau} \leftarrow \mu(d_{ij}, d_{i\tau})$

**G3.** Find the landmark with the lowest proximity
**for** $j \in \mathcal{L}$ **do**
$\rho_{j\tau} \leftarrow \nu(\delta_{1j\tau}, \delta_{2j\tau} \ldots \delta_{nj\tau})$
**if** $\rho_{j\tau} < \rho_{min}$ **then**
$\rho_{min} \leftarrow \rho_{j\tau}$
$k \leftarrow j$

**G4.** Retun the geographical location (latitude and longitude) of the landmark with the lowest proximity value
**return** $(\varphi_k, \lambda_k)$

## 3.2 Examples of distance metrics $\mu$

We list several examples of distance metrics $\mu$ that are suitable for IP geolocation.

1. Minimum delay:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = |d_{ij}^{(1)} - d_{i\tau}^{(1)}|. \tag{3.4}$$

2. $k$th-order delay:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = |d_{ij}^{(k)} - d_{i\tau}^{(k)}|. \tag{3.5}$$

   If $k = 1$, we obtain the minimum delay metric given above. If $k = m/2$ we get the median delay. Constraint-Based Geolocation [4] uses the 2.5% percentile delay metric.

3. Normalized $k$th-order delay:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \frac{|d_{ij}^{(k)} - d_{i\tau}^{(k)}|}{|d_{ij}^{(k)}| + |d_{i\tau}^{(k)}|}. \tag{3.6}$$

4. Mean sample absolute difference:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \frac{1}{m}\sum_{k=1}^{m} |d_{ij}^{(k)} - d_{i\tau}^{(k)}|. \tag{3.7}$$

5. Kullback-Leibler divergence estimate [13]:

   Let $U(x)$ denote the unit step function:

$$U(x) = \begin{cases} 0, & \text{if } x < 0, \\ \frac{1}{2}, & \text{if } x = 0, \\ 1, & \text{if } x > 0. \end{cases} \tag{3.8}$$

   The empirical cumulative distribution function (CDF) associated with $\boldsymbol{d}_{ij}$ is given by

$$P_{ij}^{E}(d) = \frac{1}{m}\sum_{k=1}^{m} U(d - d_{ij}^{(k)}), \tag{3.9}$$

   which leads to a continuous piecewise linear extension:

$$P_{ij}^{C}(d) = \begin{cases} 0, & \text{if } d < d_{ij}^{(0)}, \\ a_i d + b_i, & \text{if } d_{ij}^{(i-1)} < d < d_{ij}^{(i)}, \\ 1, & \text{if } d_{ij}^{(n+1)} < d, \end{cases} \tag{3.10}$$

   where $a_i$ and $b_i$ are chosen such that $P_{ij}^{C}(d_{ij}^{(k)}) = P_{ij}^{E}(d_{ij}^{(k)})$ for $k \in \{1 \dots m\}$, and $d_{ij}^{(0)} < d_{ij}^{(1)}$ and $d_{ij}^{(m+1)} > d_{ij}^{(m)}$ are arbitrarily chosen. In practice, we choose $d_{ij}^{(0)}$ and $d_{ij}^{(m+1)}$ such that $a_0 = a_1$, $b_0 = b_1$, respectively, $a_n = a_{n+1}$, $b_n = b_{n+1}$. The

Kullback-Leibler divergence estimator is then given by [13]

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \frac{1}{m} \sum_{k=1}^{m} \log \frac{\delta P_{ij}^{C}(d_{ij}^{(k)})}{\delta P_{i\tau}^{C}(d_{ij}^{(k)})}, \tag{3.11}$$

where $\delta P_{ij}^{C}(d_{ij}^{(k)}) = P_{ij}^{C}(d_{ij}^{(k)}) - P_{ij}^{C}(d_{ij}^{(k)} - \epsilon)$ and $0 < \epsilon < \min_k\{d_{ij}^{(k)} - d_{ij}^{(k-1)}\}$.

6. Chord distance [14–16]:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \sqrt{\sum_{k=1}^{m} \left( \frac{d_{ij}^{(k)}}{||\boldsymbol{d}_{ij}||_2} - \frac{d_{i\tau}^{(k)}}{||\boldsymbol{d}_{i\tau}||_2} \right)^2}, \tag{3.12}$$

where $||\cdot||_2$ denotes the standard Euclidean norm.

7. Chi-square distance [16]:

Let $s_{ij}^{(k)} = d_{ij}^{(k)} + d_{i\tau}^{(k)}$ and $\boldsymbol{s}_{ij} = (s_{ij}^{(k)} : 1 \leq k \leq m)$. The chi-square distance metric is then defined by

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \sqrt{||\boldsymbol{s}_{ij}||_1} \sqrt{\sum_{k=1}^{m} \frac{1}{s_{ij}^{(k)}} \left( \frac{d_{ij}^{(k)}}{||\boldsymbol{d}_{ij}||_1} - \frac{d_{i\tau}^{(k)}}{||\boldsymbol{d}_{i\tau}||_1} \right)^2}, \tag{3.13}$$

where $||\cdot||_1$ is the standard 1-norm, also known as the taxicab or Manhattan norm.

8. Distance between species profiles [16]:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \sqrt{\sum_{k=1}^{m} \left( \frac{d_{ij}^{(k)}}{||\boldsymbol{d}_{ij}||_1} - \frac{d_{i\tau}^{(k)}}{||\boldsymbol{d}_{i\tau}||_1} \right)^2}. \tag{3.14}$$

9. Hellinger distance [16, 17]:

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = \sqrt{\sum_{k=1}^{m} \left( \sqrt{\frac{d_{ij}^{(k)}}{||\boldsymbol{d}_{ij}||_1}} - \sqrt{\frac{d_{i\tau}^{(k)}}{||\boldsymbol{d}_{i\tau}||_1}} \right)^2}. \tag{3.15}$$

## 3.3 Examples of norm functions $\nu$

Let $\boldsymbol{x} \in \mathbb{R}^n$.

1. $p$-norms:

   For $0 < p < \infty$, the $p$-norm is given by

   $$\nu^p(\boldsymbol{x}) = ||\boldsymbol{x}||_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}. \tag{3.16}$$

   Note that for $0 < p < 1$, the $p$-norm is not a true norm (does not satisfy the triangle inequality). The most important $p$-norms are listed below. For $p = 2$ we get the Euclidean norm

   $$\nu^2(\boldsymbol{x}) = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{3.17}$$

   For $p = 1$ we get the Manhattan (taxicab) norm:

   $$\nu^1(\boldsymbol{x}) = \sum_{i=1}^{n} |x_i|. \tag{3.18}$$

   For $p = \infty$, the $p$-norm is the well-known Chebyshev norm:

   $$\nu^\infty(\boldsymbol{x}) = \max_{i=1,2,\ldots n} |x_i|. \tag{3.19}$$

2. Mahalanobis norm [18]:

   Let $\boldsymbol{d}_\tau^{(k)} = (d_{1\tau}^{(k)}, d_{2\tau}^{(k)}, \ldots, d_{n\tau}^{(k)})$, $\bar{d}_{i\tau} = \frac{1}{m}\sum_{k=1}^{m} d_{i\tau}^{(k)}$, and $\bar{\boldsymbol{d}}_\tau = (\bar{d}_{1\tau}, \bar{d}_{2\tau}, \ldots \bar{d}_{n\tau})$. The Mahalanobis norm is defined by

$$\nu(\boldsymbol{x}) = \sqrt{\boldsymbol{x}\mathbf{S}^{-1}\boldsymbol{x}^T} \tag{3.20}$$

   where $\mathbf{S}$ is the maximum likelihood estimate of the covariance matrix for measurements between active landmarks and target:

$$\mathbf{S} = \frac{1}{m}\sum_{k=1}^{m}(\boldsymbol{d}_\tau^{(k)} - \bar{\boldsymbol{d}}_\tau)^T(\boldsymbol{d}_\tau^{(k)} - \bar{\boldsymbol{d}}_\tau) \tag{3.21}$$

3. Normalized Euclidean norm:

   The normalized Euclidean norm is a particular case of the Mahalanobis norm, where the covariance matrix is diagonal. Using the notations for the Mahalanobis distance, we have

$$\nu(\boldsymbol{x}) = \sqrt{\sum_{i=1}^{n}\frac{x_i^2}{\sigma_i^2}}, \tag{3.22}$$

   where $\sigma_i$ is the sample standard deviation of the sample $\mathcal{S}_{i\tau}$

$$\sigma_i^2 = \frac{1}{m}\sum_{k=1}^{m}(d_{i\tau}^{(k)} - \bar{d}_{i\tau})^2. \tag{3.23}$$

## 3.4 "Named" proximity measures

By combining a distance metric with a norm according to (3.1), (3.2) and (3.3), we can obtain the proximity measures used in several well-known IP geolocation schemes.

### 3.4.1 Shortest Ping (SPing) proximity measure

The proximity measure used in Shortest Ping can be characterized by the distance metric

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = d_{i\tau}^{(1)} \text{ for all } j \in \mathcal{L}, \tag{3.24}$$

together with the norm function

$$\nu_j(\boldsymbol{x}) = \begin{cases} x_j, & \text{if } j \leq n, \\ \infty, & \text{if } n < j \leq N. \end{cases} \tag{3.25}$$

In this particular case, $\mu$ and $\nu_j$ depend only on the measurements $\boldsymbol{d}_{i\tau}$ between active landmarks and target, as no measurements $\boldsymbol{d}_{ij}$ are obtained between the landmarks themselves.

### 3.4.2 GeoPing proximity measure

The proximity measure for GeoPing consists of the distance metric

$$\mu(\boldsymbol{d}_{ij}, \boldsymbol{d}_{i\tau}) = |d_{ij}^{(1)} - d_{i\tau}^{(1)}|, \tag{3.26}$$

together with the Euclidean norm

$$\nu(\boldsymbol{x}) = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{3.27}$$

### 3.4.3 Canberra proximity measure

Canberra proximity measure is obtained by combining the normalized $k^{th}$-order delay distance metric $\mu$ with the Manhattan norm $\nu$. When there is only one measurement ($m = 1$), Canberra proximity measure is identical to Canberra distance [3, 19, 20] between the two

vectors $(d_{ij}^{(1)} : i \in \mathcal{L}_a)$ and $(d_{i\tau}^{(1)} : i \in \mathcal{L}_a)$

### 3.4.4 Clark proximity measure

Clark proximity measure is obtained by combining the normalized $k^{th}$-order delay $\mu$ with the Euclidean norm $\nu$. Like for the Canberra proximity measure, when $m = 1$ Clark proximity measure is identical to Clark distance [21]. TODO: Add formula here

### 3.4.5 Modified Clark proximity measure

Our experimental results show that using a $p$-norm with $p$ slightly larger than two (e.g., $p = 2.15$) in conjunction with the normalized delay gives better estimation results for large distances (we will later see that this choice is reducing the mean error by 4.3% and third quartile by 13.7%). We refer to this proximity measure as *Modified Clark* proximity measure.

## 3.5 Construction of Measurement Plan

We measure the landmark-landmark and landmark-target delays using three types of ping, which correspond to the following Internet protocols: Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). Because of firewalls in place, not all landmarks/targets respond to all types of ping. Generally, the PlanetLab machines always answer to the TCP ping on port 22 (SSH) because of the requirements of the PlanetLab software; however, only some of the landmarks answer to ICMP or UDP ping, because of the firewall rules at each university or company gateway, for protection against outside cyberattacks. In order to get the sample delay dataset, we take the ICMP ping measurements wherever possible, and substitute with TCP or UDP ping measurements whenever ICMP ping is blocked by firewalls.

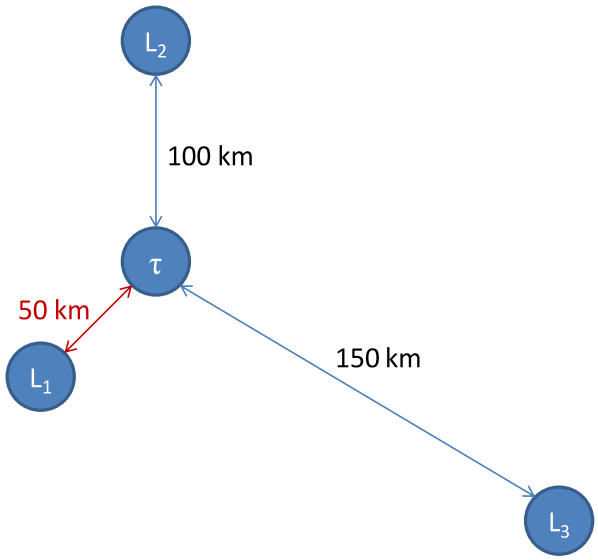


Figure 3.1: Lower bound of error (red) for pure delay-based methods.

## 3.6 Lower Bound on Error for Pure Delay-Based Algorithms

The pure delay-based class of IP geolocation algorithms estimate the target location with that of an active or passive landmark. Thus, an inherent lower bound of the geolocation error is given by the geographical distance between the target and the geographically closest landmark. For example in Fig. 3.1 with three landmarks $L_1$, $L_2$ and $L_3$ and the target $\tau$, the gelocation error of pure delay-based methods cannot be smaller than 50 km, corresponding to the geographical distance between the target ($\tau$) and the closest lanmark ($L_1$). This lower bound can however be improved by using supplementary information, e.g., geographic or demographic. As the number and density of the landmarks increases, however, the advantage gained by the supplementary information is gradually lost. By comparing the error CDF for a given proximity measure with the error CDF of the lower bound, one can get a good idea of its geolocation performance relative to the smallest possible geolocation error.

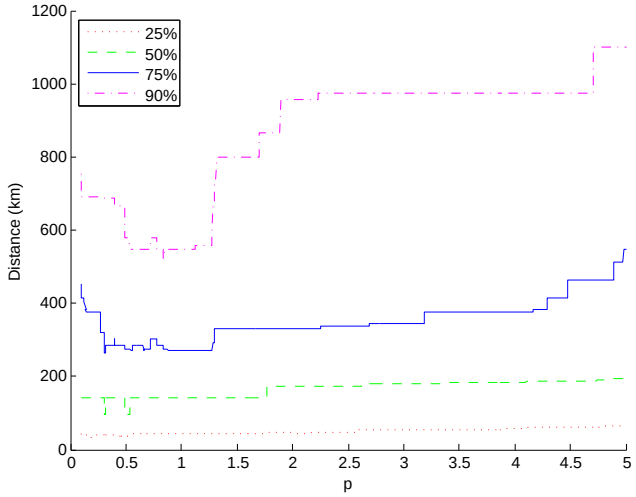


Figure 3.2: Error percentile (25%,50%,75%,90%) of the $p$-norm combined with the minimum delay difference.

## 3.7 Experimental Results

The following analysis is performed using the maximum numbers of active landmark (78 active and 3 passive). A more complete analysis, with 20, 40, 60 and 78 active landmarks out of 81 is provided in Section 3.8.

### 3.7.1 Minimum delay and $p$-norms

In the first experiment, we studied the performance resulting from the use of the minimum delay metric and the $p$-norms. We noticed in Fig. 3.2 that for very small and large values of $p$ the estimation is inefficient. A plausible explanation is that larger values of $p$ give more emphasis on landmarks which are distant from the target; these landmarks have the largest variances in measurements and contribute the most to the distance measure. A comparison of several $p$-norms and their performance with respect to the lower bound (LB) of Section 3.6 in terms of cumulative distribution function (CDF) is illustrated in Fig. 3.3. The poor performance of the infinity norm is a result of the fact that the landmarks far away from the target, which have the largest differences in terms of delays to the target, contribute little to the accuracy of geolocation. The most important statistics related to

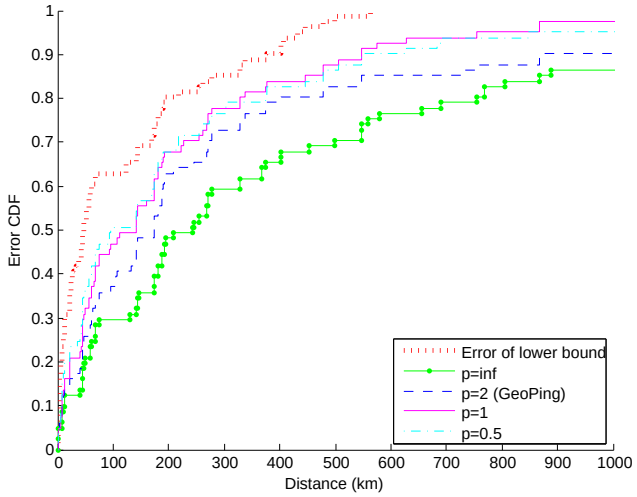


Figure 3.3: Empirical CDF of $p$-norms vs. minimum attainable error.

| Error [km] | LB | $p = 1/2$ | $p = 1$ | $p = 2$ | $p = \infty$ |
|---|---|---|---|---|---|
| mean | 118 | 222 | 212 | 421 | 560 |
| median | 48 | 95 | 141 | 173 | 243 |
| maximum | 563 | 1544 | 1544 | 4307 | 4307 |
| std. dev. | 146 | 302 | 269 | 835 | 914 |
| 1st quartile | 10 | 38 | 44 | 47 | 65 |
| 3rd quartile | 180 | 274 | 269 | 331 | 563 |

Table 3.1: Accuracy comparison of $p$-norms, for $p = \frac{1}{2}, 1, 2$ and $\infty$. with respect to the lower bound (LB)

the geolocation errors are summarized in Table 3.1.

### 3.7.2 Normalized minimum delay and $p$-norms

To achieve geolocation results that are more sensitive to smaller delay measurements, which contain the bulk of the useful geolocation information, we can use proximity measures such as Shortest Ping, Canberra, and Clark. The Canberra and Clark proximity measures are particular cases of the normalized delay metric combined with a $p$-norm when $p = 1$ and $p = 2$, respectively.

The geolocation accuracy achieved using this class of proximity measures is ilustrated

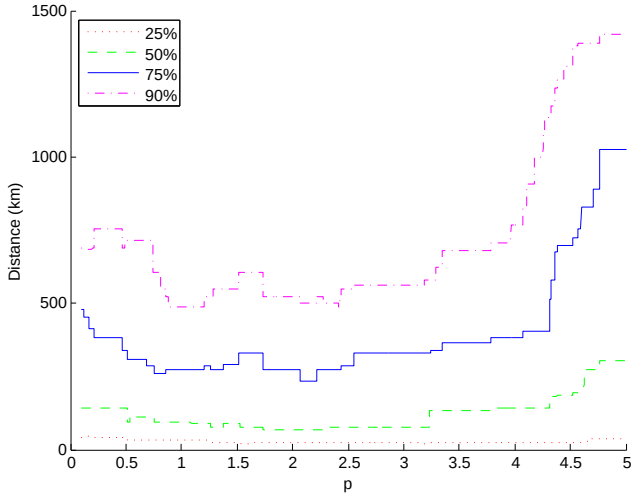


Figure 3.4: Error percentile (25%,50%,75%,90%) of the $p$-norm combined with the normalized minimum delay difference.

in Fig. 3.4. We notice that the highest accuracy is obtained when $p \approx 1$ and $p \approx 2$, and in particular, the lowest error is attained when $p \approx 2.15$. We call this proximity measure, i.e., when $\mu$ is the normalized delay difference and $\nu$ is the $p$-norm with $p = 2.15$, the *Modified Clark* method.

The empirical CDF of the error of the proximity measures for Shortest Ping, Canberra, Clark, and Modified Clark, are compared with the lower bound and Manhattan distance in Fig. 3.5. The error statistics of these methods are listed in Table 3.2. We note that Shortest Ping performs very well when the landmarks are very close to each other. As the distances among the landmarks increase, Shortest Ping is gradually outperformed by the other methods.

### 3.7.3 Minimum delay and normalized Euclidean or Mahalanobis norm

Another way of improving the accuracy is to replace the $p$-norms with norms based on the delay variances, such as in the normalized Euclidean and Mahalanobis norms. The empirical CDFs of the error using these two proximity measures are illustrated in Fig. 3.6 together with the curves for the lower bound and Shortest Ping. Unlike the previous methods, this

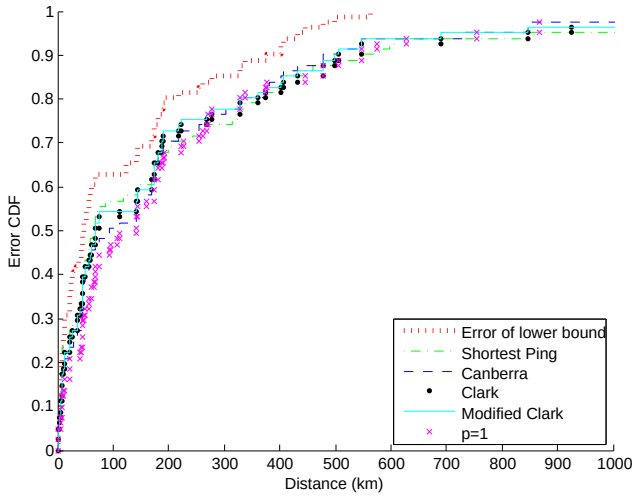


Figure 3.5: Empirical CDF for the error of $L_1$, Canberra, Clark, Modified Clark and Shortest Ping distance vs. minimum attainable error.

| Error [km] | LB | $L_1$ | Canberra | Clark | M. Clark | SPing |
|---|---|---|---|---|---|---|
| mean | 118 | 212 | 203 | 210 | 201 | 270 |
| median | 48 | 141 | 93 | 67 | 67 | 66 |
| maximum | 563 | 1544 | 1874 | 1874 | 1874 | 3918 |
| std. dev. | 146 | 269 | 290 | 313 | 302 | 605 |
| 1st quartile | 10 | 44 | 33 | 22 | 22 | 13 |
| 3rd quartile | 180 | 269 | 271 | 271 | 234 | 317 |

Table 3.2: Accuracy comparison of $L_1$, Canberra, Clark, Modified Clark and Shortest-ping distances vs. minimum attainable error.

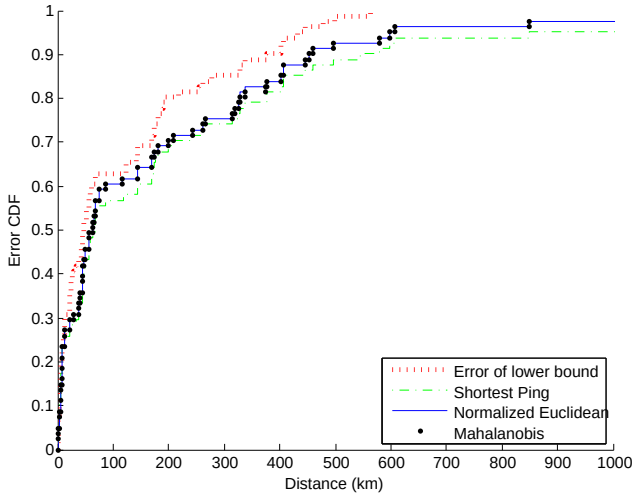


Figure 3.6: Empirical CDF for the error of Shortest Ping and minimum delay with normalized Euclidean and Mahalanobis norms vs. minimum attainable error.

| Error [km] | LB | SPing | Mah | NE |
|---|---|---|---|---|
| mean | 118 | 270 | 209 | 209 |
| median | 48 | 66 | 62 | 62 |
| maximum | 563 | 3918 | 3918 | 3918 |
| std. dev. | 146 | 605 | 469 | 468 |
| 1st quartile | 10 | 13 | 13 | 13 |
| 3rd quartile | 180 | 317 | 277 | 277 |

Table 3.3: Accuracy comparison of Shortest Ping and minimum delay difference with Mahalanobis and normalized Euclidean norms vs. minimum attainable error.

method performs consistently better than Shortest Ping over all distances. Error statistics for this comparison are given in Table 3.3.

### 3.7.4 Kullback-Leibler divergence and $p$-norms proximity measures

The results of the next experiment, using Kullback-Leibler divergence with $p$-norms, are shown in Fig. 3.7. From this figure we can conclude that the Kullback-Leibler divergence gives worse results than the minimum delay. Similar results were obtained for other distance metrics, i.e., chord distance, chi-square distance, distance between species profiles, and

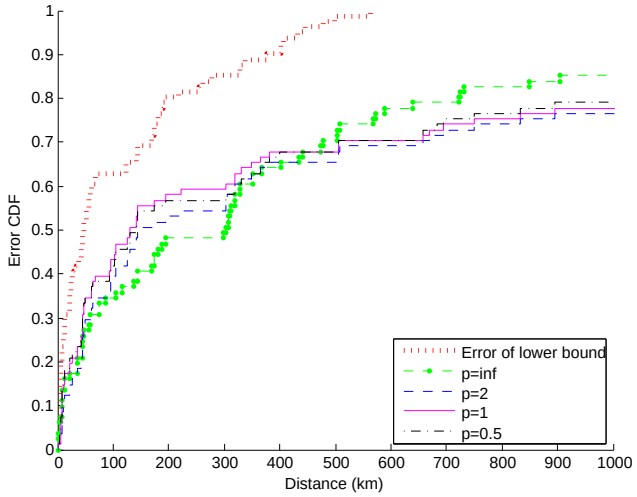


Figure 3.7: Empirical CDF for the error of $p$-norms with empirical Kullback-Leibler divergence estimate.

Hellinger distance.

## 3.8 Comprehesive Empirical Study

In order to determine the best combination of $\mu$ and $\nu$, we performed a comprehensive empirical study with the following setup.

Choices of $\mu$:

- Absolute difference of minimum delays (Min)
- Absolute difference of first quartiles - 25% percentiles (Q1)
- Absolute difference of medians (Median)
- Normalized absolute difference of minimum delays (Norml Min)
- Normalized absolute difference of first quartiles (Norml Q1)
- Normalized absolute difference of medians (Norml Median)
- Mean sample absolute difference (Abs Diff)

- Kullback-Leibler divergence estimate from samples (KL)
- Chord distance (Chord)
- Chi-square distance (Chi-Square)
- Distance between species profiles (Sp. profiles)
- Hellinger distance between samples

Choices of $\nu$:

- $p$-norm with $p = 1/2$
- Manhattan norm - $p$-norm with $p = 1$
- Euclidean norm - $p$-norm with $p = 2$
- Chebyshev norm - $p$-norm with $p = \infty$
- Mahalanobis norm (Mah)
- Normalized Euclidean (NE)

Choices of active and passive landmarks:

- 20 active landmarks and 61 passive landmarks
- 40 active landmarks and 41 passive landmarks
- 60 active landmarks and 21 passive landmarks
- 78 active landmarks and 3 passive landmarks

Choices of statistics considered:

- Mean error
- Median error

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 245 | 348 | 389 | 576 | 351 | 328 |
| Q1 | 370 | 347 | 444 | 528 | 245 | 289 |
| Median | 378 | 359 | 429 | 496 | 248 | 273 |
| Norml Min | 252 | 267 | 301 | 930 | 796 | 797 |
| Norml Q1 | 376 | 381 | 425 | 929 | 848 | 896 |
| Norml Median | 411 | 408 | 432 | 918 | 886 | 892 |
| Abs Diff | 384 | 403 | 540 | 625 | 279 | 289 |
| KL | 834 | 945 | 1041 | 1566 | 1266 | 1090 |
| Chord | 1241 | 1082 | 1042 | 999 | 1771 | 1643 |
| Chi-Square | 1193 | 1069 | 1084 | 1019 | 1645 | 1672 |
| Sp. Profiles | 1193 | 1085 | 1023 | 964 | 1730 | 1636 |
| Hellinger | 1230 | 1094 | 1078 | 1094 | 1652 | 1679 |

Table 3.4: Mean error for 20 active landmarks and 61 passive landmarks

- Maximum error
- First quartile of error (25% percentile of error)
- Median of error
- Third quartile of error (75% percentile of error)

Tables 3.4, 3.5, 3.6 and 3.7 we can observe that the mean error is consistently under 300 km when:

- $\mu$ is the absolute difference of the minimum delay and $\nu$ is the $p$-norm with $p = 1/2$;
- $\mu$ is the normalized absolute difference of the minimum delay and $\nu$ is the $p$-norm with either $p = 1/2$ or $p = 1$;
- $\mu$ is the absolute value difference of the first quartile, median or the mean sample absolute difference and $\nu$ is the Mahalanobis distance.

We remark that the Mahalanobis and normalized Euclidean norms result in more accurate geolocation than other choices of $\nu$ for the case of 78 active landmarks, the error being about 25% smaller than the minimum error of the alternatives.

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 220 | 234 | 423 | 589 | 290 | 314 |
| Q1 | 376 | 382 | 496 | 519 | 215 | 337 |
| Median | 395 | 397 | 475 | 472 | 208 | 336 |
| Norml Min | 207 | 210 | 253 | 1205 | 1056 | 1031 |
| Norml Q1 | 337 | 339 | 357 | 1180 | 1134 | 1155 |
| Norml Median | 348 | 347 | 372 | 1181 | 1137 | 1146 |
| Abs Diff | 396 | 426 | 516 | 628 | 233 | 353 |
| KL | 958 | 1007 | 1133 | 1599 | 1442 | 1316 |
| Chord | 1124 | 948 | 853 | 958 | 2248 | 2178 |
| Chi-Square | 1113 | 950 | 950 | 939 | 2093 | 2151 |
| Sp. Profiles | 1144 | 952 | 857 | 910 | 2237 | 2194 |
| Hellinger | 1153 | 1052 | 855 | 1019 | 2267 | 2278 |

Table 3.5: Mean error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 217 | 208 | 411 | 578 | 289 | 278 |
| Q1 | 378 | 380 | 526 | 800 | 219 | 261 |
| Median | 371 | 378 | 440 | 597 | 200 | 255 |
| Norml Min | 217 | 183 | 199 | 1348 | 1263 | 1257 |
| Norml Q1 | 343 | 283 | 323 | 1309 | 1301 | 1310 |
| Norml Median | 347 | 343 | 348 | 1258 | 1295 | 1307 |
| Abs Diff | 351 | 408 | 504 | 737 | 209 | 261 |
| KL | 642 | 678 | 781 | 2448 | 1615 | 1580 |
| Chord | 1009 | 865 | 847 | 978 | 2617 | 2595 |
| Chi-Square | 1001 | 891 | 799 | 842 | 2541 | 2605 |
| Sp. Profiles | 1022 | 898 | 831 | 838 | 2635 | 2575 |
| Hellinger | 1065 | 922 | 822 | 851 | 2618 | 2625 |

Table 3.6: Mean error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 222 | 213 | 422 | 561 | 209 | 209 |
| Q1 | 374 | 389 | 472 | 804 | 181 | 185 |
| Median | 371 | 385 | 413 | 595 | 168 | 180 |
| Norml Min | 254 | 203 | 211 | 1528 | 1457 | 1458 |
| Norml Q1 | 354 | 309 | 302 | 1568 | 1457 | 1466 |
| Norml Median | 373 | 340 | 339 | 1568 | 1471 | 1470 |
| Abs Diff | 373 | 419 | 481 | 742 | 171 | 187 |
| KL | 623 | 658 | 781 | 2125 | 1856 | 1778 |
| Chord | 1147 | 979 | 922 | 1013 | 3026 | 3005 |
| Chi-Square | 998 | 928 | 863 | 917 | 3058 | 3006 |
| Sp. Profiles | 1151 | 951 | 889 | 949 | 3035 | 2969 |
| Hellinger | 1151 | 954 | 933 | 937 | 3010 | 3006 |

Table 3.7: Mean error for 78 active landmarks and 3 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 283 | 616 | 754 | 915 | 610 | 606 |
| Q1 | 599 | 552 | 831 | 860 | 276 | 501 |
| Median | 610 | 574 | 728 | 751 | 274 | 492 |
| Norml Min | 355 | 394 | 413 | 1279 | 1213 | 1216 |
| Norml Q1 | 706 | 722 | 744 | 1306 | 1218 | 1314 |
| Norml Median | 724 | 726 | 761 | 1308 | 1228 | 1317 |
| Abs Diff | 575 | 629 | 907 | 1011 | 323 | 498 |
| KL | 1124 | 1218 | 1270 | 1434 | 1276 | 1215 |
| Chord | 1055 | 1067 | 951 | 837 | 1348 | 1286 |
| Chi-Square | 1062 | 1017 | 1020 | 924 | 1326 | 1294 |
| Sp. Profiles | 985 | 1018 | 885 | 903 | 1372 | 1282 |
| Hellinger | 1074 | 1010 | 986 | 1020 | 1320 | 1322 |

Table 3.8: Standard deviation of error for 20 active landmarks and 61 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 279 | 281 | 828 | 920 | 582 | 593 |
| Q1 | 684 | 689 | 919 | 856 | 219 | 650 |
| Median | 705 | 716 | 844 | 751 | 218 | 656 |
| Norml Min | 268 | 267 | 313 | 1330 | 1258 | 1264 |
| Norml Q1 | 678 | 682 | 681 | 1313 | 1257 | 1331 |
| Norml Median | 678 | 683 | 691 | 1314 | 1253 | 1335 |
| Abs Diff | 690 | 706 | 833 | 1016 | 286 | 664 |
| KL | 1174 | 1222 | 1305 | 1406 | 1323 | 1299 |
| Chord | 847 | 786 | 730 | 887 | 1395 | 1330 |
| Chi-Square | 934 | 865 | 928 | 911 | 1455 | 1394 |
| Sp. Profiles | 912 | 787 | 754 | 843 | 1396 | 1332 |
| Hellinger | 958 | 958 | 754 | 910 | 1441 | 1326 |

Table 3.9: Standard deviation of error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 264 | 236 | 828 | 919 | 591 | 592 |
| Q1 | 686 | 694 | 851 | 983 | 262 | 507 |
| Median | 680 | 699 | 776 | 736 | 230 | 505 |
| Norml Min | 273 | 234 | 246 | 1387 | 1346 | 1354 |
| Norml Q1 | 678 | 660 | 682 | 1378 | 1350 | 1381 |
| Norml Median | 679 | 683 | 689 | 1349 | 1332 | 1383 |
| Abs Diff | 682 | 711 | 836 | 1070 | 236 | 506 |
| KL | 704 | 813 | 971 | 1556 | 1416 | 1441 |
| Chord | 854 | 776 | 826 | 895 | 1277 | 1238 |
| Chi-Square | 918 | 848 | 897 | 956 | 1325 | 1277 |
| Sp. Profiles | 856 | 772 | 826 | 754 | 1274 | 1266 |
| Hellinger | 925 | 855 | 828 | 785 | 1284 | 1248 |

Table 3.10: Standard deviation of error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 302 | 270 | 835 | 914 | 469 | 468 |
| Q1 | 686 | 699 | 791 | 986 | 241 | 241 |
| Median | 688 | 706 | 702 | 735 | 213 | 235 |
| Norml Min | 347 | 290 | 313 | 1338 | 1358 | 1358 |
| Norml Q1 | 687 | 677 | 679 | 1376 | 1358 | 1351 |
| Norml Median | 693 | 682 | 685 | 1376 | 1349 | 1349 |
| Abs Diff | 699 | 713 | 812 | 1070 | 212 | 234 |
| KL | 707 | 740 | 993 | 1474 | 1499 | 1497 |
| Chord | 924 | 870 | 817 | 918 | 1008 | 1006 |
| Chi-Square | 971 | 882 | 922 | 1031 | 982 | 1042 |
| Sp. Profiles | 872 | 829 | 801 | 892 | 987 | 1060 |
| Hellinger | 974 | 907 | 930 | 855 | 1034 | 1042 |

Table 3.11: Standard deviation of error for 78 active landmarks and 3 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 1260 | 3945 | 4307 | 4307 | 3918 | 3918 |
| Q1 | 4149 | 4149 | 4149 | 4149 | 1233 | 4149 |
| Median | 4149 | 4149 | 4149 | 4149 | 1233 | 4149 |
| Norml Min | 2352 | 2352 | 2352 | 4307 | 4307 | 4307 |
| Norml Q1 | 4149 | 4149 | 4149 | 4307 | 4307 | 4307 |
| Norml Median | 4149 | 4149 | 4149 | 4307 | 4307 | 4307 |
| Abs Diff | 4149 | 4149 | 4149 | 4149 | 1883 | 4149 |
| KL | 4149 | 4149 | 4149 | 4149 | 4137 | 4138 |
| Chord | 4109 | 4296 | 4296 | 4296 | 4264 | 4264 |
| Chi-Square | 4149 | 4149 | 4296 | 4296 | 4264 | 4264 |
| Sp. Profiles | 4089 | 4296 | 4296 | 4195 | 4264 | 4264 |
| Hellinger | 4149 | 4149 | 4296 | 4314 | 4264 | 4264 |

Table 3.12: Maximum error for 20 active landmarks and 61 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 1376 | 1376 | 4307 | 4307 | 3918 | 3918 |
| Q1 | 4149 | 4149 | 4149 | 4149 | 1130 | 4149 |
| Median | 4149 | 4149 | 4149 | 4149 | 1130 | 4149 |
| Norml Min | 1218 | 1218 | 1376 | 4307 | 4307 | 4307 |
| Norml Q1 | 4149 | 4149 | 4149 | 4307 | 4307 | 4307 |
| Norml Median | 4149 | 4149 | 4149 | 4307 | 4307 | 4307 |
| Abs Diff | 4149 | 4149 | 4149 | 4149 | 1883 | 4149 |
| KL | 4149 | 4149 | 4149 | 4149 | 4149 | 4149 |
| Chord | 3877 | 3877 | 3877 | 4089 | 4313 | 4264 |
| Chi-Square | 4149 | 4149 | 4149 | 4149 | 4313 | 4264 |
| Sp. Profiles | 4288 | 3877 | 4296 | 4089 | 4313 | 4264 |
| Hellinger | 4149 | 4149 | 4149 | 4296 | 4313 | 4264 |

Table 3.13: Maximum error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 1146 | 1056 | 4307 | 4307 | 3918 | 3918 |
| Q1 | 4149 | 4149 | 4149 | 4149 | 1188 | 4149 |
| Median | 4149 | 4149 | 4149 | 4149 | 1130 | 4149 |
| Norml Min | 1146 | 1130 | 1143 | 4325 | 4308 | 4308 |
| Norml Q1 | 4149 | 4149 | 4149 | 4308 | 4308 | 4308 |
| Norml Median | 4149 | 4149 | 4149 | 4308 | 4308 | 4308 |
| Abs Diff | 4149 | 4149 | 4149 | 4149 | 1130 | 4149 |
| KL | 3186 | 3842 | 4120 | 4328 | 4258 | 4258 |
| Chord | 3877 | 3877 | 3896 | 4108 | 4264 | 4264 |
| Chi-Square | 4149 | 4149 | 4149 | 4312 | 4264 | 4264 |
| Sp. Profiles | 3877 | 3877 | 3896 | 4011 | 4264 | 4264 |
| Hellinger | 4149 | 4149 | 4149 | 4328 | 4264 | 4264 |

Table 3.14: Maximum error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 1544 | 1544 | 4307 | 4307 | 3918 | 3918 |
| Q1 | 4149 | 4149 | 4149 | 4149 | 1188 | 1188 |
| Median | 4149 | 4149 | 4149 | 4149 | 1130 | 1130 |
| Norml Min | 1875 | 1875 | 1875 | 4321 | 4321 | 4321 |
| Norml Q1 | 4149 | 4149 | 4149 | 4321 | 4321 | 4321 |
| Norml Median | 4149 | 4149 | 4149 | 4321 | 4321 | 4321 |
| Abs Diff | 4149 | 4149 | 4149 | 4149 | 1130 | 1130 |
| KL | 3186 | 3186 | 4120 | 4328 | 4325 | 4325 |
| Chord | 3877 | 3877 | 3877 | 4326 | 4294 | 4264 |
| Chi-Square | 4149 | 4149 | 4149 | 4312 | 4264 | 4264 |
| Sp. Profiles | 3877 | 3877 | 3877 | 4089 | 4328 | 4264 |
| Hellinger | 4149 | 4149 | 4149 | 4328 | 4264 | 4264 |

Table 3.15: Maximum error for 78 active landmarks and 3 passive landmarks

From Tables 3.8, 3.9, 3.10 and 3.11, we find that the best combinations of $\mu$ and $\nu$, namely $p$-norm with $p = 1/2$ in combination with the absolute difference of the minimum delay and Mahalanobis norm in combination with either the absolute difference of the first quartile or the absolute difference of the median, have also the lowest variance. Tables 3.12, 3.13, 3.14 and 3.15 show that the maximum error is quite large, most likely because of the incorrect locations reported by a couple of the landmarks. However, the estimators which performed well with respect to the other statistics also have lower values for the maximum error.

From Tables 3.16, 3.17, 3.18 and 3.19 we can identify two algorithms which provide a low first quartile of estimation error: Clark proximity measure (which is Euclidean norm combined with normalized minimum, first quartile or median normalized absolute difference), and, to a lesser extent, Canberra proximity measure. The notable exception is the case of 78 active landmarks, when Mahalanobis and normalized Euclidean distance are combined with minimum, first quartile, median or mean sample absolute difference, resulting in an 25% quantile error of only 13 km, the same as with the Shortest Ping algorithm, and only 3 km larger than the lower bound (10 km).

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 42 | 44 | 44 | 64 | 45 | 44 |
| Q1 | 44 | 44 | 44 | 73 | 44 | 44 |
| Median | 44 | 42 | 44 | 96 | 46 | 40 |
| Norml Min | 40 | 44 | 44 | 58 | 56 | 51 |
| Norml Q1 | 44 | 42 | 44 | 58 | 67 | 65 |
| Norml Median | 44 | 44 | 45 | 71 | 91 | 58 |
| Abs Diff | 44 | 47 | 62 | 56 | 45 | 44 |
| KL | 131 | 131 | 143 | 271 | 172 | 143 |
| Chord | 420 | 338 | 420 | 371 | 547 | 482 |
| Chi-Square | 395 | 367 | 411 | 367 | 422 | 482 |
| Sp. Profiles | 411 | 367 | 420 | 351 | 414 | 482 |
| Hellinger | 395 | 374 | 420 | 364 | 449 | 454 |

Table 3.16: First quartile error for 20 active landmarks and 61 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 42 | 44 | 45 | 60 | 44 | 45 |
| Q1 | 42 | 44 | 45 | 68 | 44 | 45 |
| Median | 42 | 42 | 44 | 60 | 43 | 45 |
| Norml Min | 25 | 35 | 44 | 141 | 67 | 58 |
| Norml Q1 | 42 | 35 | 44 | 145 | 105 | 98 |
| Norml Median | 39 | 42 | 45 | 141 | 105 | 73 |
| Abs Diff | 44 | 47 | 62 | 73 | 44 | 46 |
| KL | 143 | 169 | 198 | 367 | 367 | 303 |
| Chord | 504 | 360 | 311 | 347 | 1044 | 1060 |
| Chi-Square | 462 | 374 | 340 | 285 | 745 | 936 |
| Sp. Profiles | 504 | 360 | 383 | 413 | 967 | 1089 |
| Hellinger | 462 | 418 | 302 | 410 | 935 | 1270 |

Table 3.17: First quartile error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 42 | 44 | 46 | 60 | 41 | 40 |
| Q1 | 44 | 45 | 73 | 217 | 41 | 40 |
| Median | 42 | 37 | 41 | 124 | 39 | 37 |
| Norml Min | 35 | 33 | 43 | 193 | 162 | 65 |
| Norml Q1 | 44 | 35 | 35 | 185 | 189 | 161 |
| Norml Median | 35 | 39 | 35 | 161 | 192 | 126 |
| Abs Diff | 44 | 46 | 49 | 119 | 44 | 39 |
| KL | 125 | 125 | 178 | 722 | 468 | 359 |
| Chord | 394 | 297 | 276 | 312 | 1510 | 1515 |
| Chi-Square | 333 | 297 | 226 | 246 | 1315 | 1529 |
| Sp. Profiles | 433 | 311 | 268 | 307 | 1542 | 1515 |
| Hellinger | 387 | 300 | 277 | 303 | 1528 | 1529 |

Table 3.18: First quartile error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 38 | 44 | 48 | 65 | 13 | 13 |
| Q1 | 44 | 44 | 61 | 217 | 13 | 13 |
| Median | 39 | 39 | 41 | 124 | 13 | 13 |
| Norml Min | 40 | 33 | 22 | 416 | 330 | 330 |
| Norml Q1 | 40 | 35 | 25 | 416 | 330 | 343 |
| Norml Median | 35 | 32 | 21 | 416 | 359 | 343 |
| Abs Diff | 44 | 46 | 49 | 124 | 13 | 13 |
| KL | 125 | 125 | 125 | 870 | 472 | 363 |
| Chord | 477 | 354 | 297 | 314 | 2315 | 2315 |
| Chi-Square | 276 | 302 | 218 | 209 | 2324 | 2315 |
| Sp. Profiles | 474 | 354 | 277 | 371 | 2315 | 2197 |
| Hellinger | 443 | 302 | 283 | 289 | 2315 | 2315 |

Table 3.19: First quartile error for 78 active landmarks and 3 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 142 | 174 | 174 | 255 | 181 | 143 |
| Q1 | 174 | 181 | 181 | 199 | 168 | 173 |
| Median | 181 | 181 | 191 | 255 | 173 | 173 |
| Norml Min | 143 | 143 | 143 | 288 | 191 | 191 |
| Norml Q1 | 168 | 143 | 205 | 299 | 295 | 302 |
| Norml Median | 181 | 174 | 193 | 288 | 295 | 227 |
| Abs Diff | 188 | 188 | 193 | 235 | 174 | 174 |
| KL | 326 | 349 | 368 | 1186 | 893 | 679 |
| Chord | 888 | 814 | 801 | 788 | 1497 | 1345 |
| Chi-Square | 822 | 814 | 814 | 718 | 1334 | 1477 |
| Sp. Profiles | 925 | 814 | 801 | 706 | 1497 | 1345 |
| Hellinger | 860 | 822 | 814 | 758 | 1334 | 1477 |

Table 3.20: Median error for 20 active landmarks and 61 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 142 | 143 | 174 | 255 | 143 | 168 |
| Q1 | 143 | 173 | 174 | 188 | 168 | 168 |
| Median | 173 | 174 | 181 | 245 | 143 | 174 |
| Norml Min | 93 | 141 | 142 | 657 | 402 | 362 |
| Norml Q1 | 142 | 142 | 143 | 534 | 534 | 534 |
| Norml Median | 143 | 143 | 143 | 629 | 657 | 534 |
| Abs Diff | 173 | 181 | 188 | 222 | 168 | 174 |
| KL | 376 | 376 | 507 | 1186 | 1031 | 867 |
| Chord | 830 | 758 | 691 | 718 | 2399 | 2374 |
| Chi-Square | 822 | 714 | 691 | 718 | 1884 | 2325 |
| Sp. Profiles | 830 | 758 | 691 | 684 | 2399 | 2374 |
| Hellinger | 822 | 798 | 718 | 758 | 2506 | 2444 |

Table 3.21: Median error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 142 | 142 | 173 | 209 | 116 | 75 |
| Q1 | 168 | 174 | 222 | 423 | 142 | 142 |
| Median | 168 | 173 | 181 | 388 | 142 | 118 |
| Norml Min | 96 | 67 | 116 | 876 | 764 | 764 |
| Norml Q1 | 143 | 92 | 132 | 764 | 832 | 876 |
| Norml Median | 143 | 143 | 168 | 657 | 876 | 876 |
| Abs Diff | 168 | 174 | 188 | 288 | 142 | 142 |
| KL | 376 | 365 | 376 | 2980 | 1186 | 1148 |
| Chord | 792 | 578 | 571 | 718 | 3041 | 2946 |
| Chi-Square | 792 | 665 | 490 | 578 | 3027 | 3027 |
| Sp. Profiles | 792 | 670 | 509 | 653 | 3041 | 2946 |
| Hellinger | 828 | 691 | 665 | 677 | 3041 | 3027 |

Table 3.22: Median error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 96 | 141 | 173 | 244 | 62 | 62 |
| Q1 | 168 | 174 | 199 | 402 | 64 | 67 |
| Median | 143 | 173 | 181 | 364 | 67 | 68 |
| Norml Min | 141 | 93 | 68 | 1313 | 989 | 989 |
| Norml Q1 | 142 | 112 | 75 | 1313 | 989 | 989 |
| Norml Median | 143 | 142 | 142 | 1313 | 1032 | 1014 |
| Abs Diff | 168 | 181 | 188 | 347 | 68 | 75 |
| KL | 351 | 354 | 351 | 1746 | 1301 | 1189 |
| Chord | 860 | 756 | 674 | 893 | 3329 | 3288 |
| Chi-Square | 689 | 670 | 485 | 514 | 3354 | 3329 |
| Sp. Profiles | 941 | 670 | 677 | 668 | 3329 | 3288 |
| Hellinger | 860 | 674 | 663 | 712 | 3329 | 3329 |

Table 3.23: Median error for 78 active landmarks and 3 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 377 | 406 | 377 | 732 | 383 | 402 |
| Q1 | 399 | 446 | 378 | 560 | 338 | 383 |
| Median | 378 | 399 | 378 | 509 | 332 | 375 |
| Norml Min | 374 | 333 | 373 | 1328 | 939 | 841 |
| Norml Q1 | 384 | 377 | 427 | 1328 | 1100 | 1328 |
| Norml Median | 394 | 446 | 394 | 1328 | 1236 | 1328 |
| Abs Diff | 469 | 446 | 494 | 586 | 402 | 383 |
| KL | 1059 | 1232 | 1406 | 2666 | 1702 | 1406 |
| Chord | 1653 | 1267 | 1231 | 1272 | 3064 | 2735 |
| Chi-Square | 1616 | 1376 | 1272 | 1205 | 2804 | 2634 |
| Sp. Profiles | 1616 | 1342 | 1267 | 1177 | 2949 | 2583 |
| Hellinger | 1653 | 1411 | 1267 | 1344 | 2804 | 2801 |

Table 3.24: Third quartile error for 20 active landmarks and 61 passive landmarks

From Tables 3.20, 3.21, 3.22, and 3.23 we note that for a small percentage of active landmarks (20, 40 and 60 active landmarks) several methods are comparable: minimum absolute delay difference with $p$-norm ($p = 1/2$), normalized minimum absolute delay difference with $p$-norms ($p = 1$ or $p = 2$), and minimum absolute delay difference or first quartile absolute delay difference in combination with the normalized Euclidean norm. For a large percentage of active landmarks (78 active landmarks), minimum absolute delay difference and first quartile absolute delay difference in combination with either Mahalanobis distance or normalized Euclidean distance dominate the rest of the methods.

From Tables 3.24, 3.25, 3.26, and 3.27 we conclude that several of the measures give similar results, such as minimum absolute delay difference in combination with $p$-norms (for $p = 1/2$, $p = 1$, or $p = 2$), Mahalanobis distance or normalized Euclidean. Similar results are yielded by the normalized minimum absolute delay difference in combination with $p$-norms for $p = 1$ or $p = 2$. The Kullback-Leibler divergence, chord, chi-square, species profiles, and Hellinger distance metrics resulted in relatively poor geolocation performance.

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 303 | 329 | 369 | 707 | 333 | 332 |
| Q1 | 375 | 375 | 383 | 563 | 333 | 383 |
| Median | 332 | 332 | 394 | 514 | 331 | 383 |
| Norml Min | 303 | 303 | 329 | 1752 | 1490 | 1490 |
| Norml Q1 | 341 | 329 | 330 | 1567 | 1567 | 1567 |
| Norml Median | 342 | 330 | 331 | 1567 | 1527 | 1567 |
| Abs Diff | 499 | 499 | 672 | 586 | 344 | 416 |
| KL | 1270 | 1321 | 1845 | 2666 | 2325 | 2248 |
| Chord | 1539 | 1422 | 1108 | 1139 | 3583 | 3305 |
| Chi-Square | 1495 | 1255 | 1127 | 1160 | 3545 | 3444 |
| Sp. Profiles | 1539 | 1422 | 1094 | 1132 | 3583 | 3370 |
| Hellinger | 1551 | 1376 | 1117 | 1255 | 3664 | 3528 |

Table 3.25: Third quartile error for 40 active landmarks and 41 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 303 | 272 | 338 | 707 | 332 | 320 |
| Q1 | 394 | 375 | 629 | 1120 | 327 | 329 |
| Median | 394 | 330 | 394 | 806 | 315 | 329 |
| Norml Min | 310 | 263 | 284 | 2420 | 1933 | 1933 |
| Norml Q1 | 340 | 284 | 327 | 2147 | 2147 | 2147 |
| Norml Median | 342 | 331 | 330 | 1933 | 1933 | 2147 |
| Abs Diff | 357 | 446 | 670 | 914 | 320 | 332 |
| KL | 1033 | 1033 | 1063 | 3834 | 2680 | 2680 |
| Chord | 1393 | 1231 | 1078 | 1285 | 3791 | 3755 |
| Chi-Square | 1372 | 1220 | 1060 | 1130 | 3791 | 3780 |
| Sp. Profiles | 1487 | 1288 | 1041 | 1154 | 3791 | 3755 |
| Hellinger | 1428 | 1231 | 1108 | 1130 | 3791 | 3780 |

Table 3.26: Third quartile error for 60 active landmarks and 21 passive landmarks

| Error [km] | p=1/2 | p=1 | p=2 | p=$\infty$ | Mah | NE |
|---|---|---|---|---|---|---|
| Min | 274 | 270 | 331 | 563 | 277 | 277 |
| Q1 | 377 | 377 | 427 | 1120 | 277 | 317 |
| Median | 394 | 330 | 394 | 795 | 262 | 277 |
| Norml Min | 340 | 272 | 272 | 2482 | 2420 | 2420 |
| Norml Q1 | 392 | 284 | 286 | 2515 | 2420 | 2420 |
| Norml Median | 446 | 341 | 341 | 2515 | 2420 | 2420 |
| Abs Diff | 416 | 459 | 446 | 875 | 261 | 307 |
| KL | 842 | 1056 | 1088 | 3842 | 3446 | 3426 |
| Chord | 1605 | 1231 | 1298 | 1429 | 3842 | 3826 |
| Chi-Square | 1490 | 1220 | 1114 | 1150 | 3849 | 3842 |
| Sp. Profiles | 1620 | 1231 | 1220 | 1311 | 3842 | 3826 |
| Hellinger | 1604 | 1243 | 1377 | 1230 | 3842 | 3842 |

Table 3.27: Third quartile error for 78 active landmarks and 3 passive landmarks

## 3.9 Analysis and Interpretation

The accuracy of the pure delay-based class of IP geolocation algorithms is mainly impacted by the fact that not all the active landmarks give the same amount of information about the target location. Landmarks that are close to the target and have a relatively direct connection to it via few intermediate hops provide most of the useful information for target localization. By contrast, the active landmarks which are far away from the target, or connected to the target via a circuitous path are of little or no value in localizing the target location.

The main challenge in improving the precision of geolocation consists in choosing $\mu$ and $\nu$ to penalize the active landmarks that provide little value for geolocation. Our research has led to the discovery of two successful ways of achieving a proper weighting of the information. One way of applying the penalty is to use the normalized delay distance $\frac{|x-y|}{|x|+|y|}$. Thus, when one of the delays is large as a result of large distance between the active landmark and target and/or circuitous path, its contribution to the objective function is scaled down. Conversely, when both $x$ and $y$ are small, the influence of the delay difference over the objective function is emphasized. The normalized delay distance can be combined

with different $p$-norms, distinguishable cases being $p = 1$, which gives Canberra distance, and $p = 2$ which gives Clark's distance.

The second way of weighting the measurements is to use the sample variance/covariance as a penalty for the unreliable landmarks. This is based on the empirical observation that the reliability of the active landmarks is negatively correlated with the sample variance/covariance of multiple measurements from the active landmarks to the target. Consequently, a large variance indicates that either a landmark is far away from the target or connected to it via a circuitous path, since multiple intermediate hops have a positive contribution to the variance. Conversely, a small variance suggests that the information from an active landmark is reliable and therefore its contribution to the overall geolocation procedure should be enhanced. This is the case for the Mahalanobis and normalized Euclidean distance metrics, which provide improved geolocation accuracy compared to the earlier methods. TODO: explain about influence of passive/active landmarks ratio

# Chapter 4: Statistical Geolocation of Internet Hosts

In this chapter we develop a statistical approach to IP geolocation. Our approach consists of several steps. First, a "profile" of each landmark is constructed using the distance-delay pairs amongst the landmarks, resulting in a scatterplot for each landmark. Second, the joint probability distribution of the distance and delay is approximated using bivariate kernel density estimation. A Gaussian kernel is used for density estimation. Finally, a force-directed algorithm is used to obtain an estimate of the target location.

## 4.1 Construction of Landmark Profiles

The profile of an active landmark $L_i$, $i \in \mathcal{L}_a$ consists of the set of all distance-delay pair measurements originating at $L_i$ towards the other (active or passive) landmarks $L_j$, where[1] $j \in \mathcal{L} \backslash \{i\}$. Our construction of landmark profiles is similar to that of Gueye *et al.* [5]. Multiple measurements are obtained between every pair of landmarks at different times, yielding different delays (the distance between each pair of landmarks remains constant, as

Figure 4.1: Landmark distribution over the continental U.S.

[1]Here, $A \backslash B \triangleq A \cap B^c$, where $B^c$ denotes the complement of the set $B$.

landmarks are not mobile). In our experimental study, we used 85 servers in the PlanetLab research network [22]. The server locations are shown in Fig. 4.1. We obtained RTT measurements using the *ping* utility five times every 15 minutes for a period of one week, yielding up to $M = 282,240$ measurements for each target (in practice, not all measurements are successful). From the measurements, we obtained a scatterplot for each active landmark $L_i$, $i \in \mathcal{L}_a$ by taking delay measurements from $L_i$ to all other landmarks (see Fig. 4.2).

For clarity of presentation, the scatterplot in Fig. 4.2 shows only the minimum delay measurements between *planet1.cs.stanford.edu* and 79 other PlanetLab nodes. The SPing and GeoPing methods use only minimum delay measurements. The CBG scheme uses the 2.5 percentile of measurements. By contrast, the statistical geolocation scheme proposed in this thesis uses all of the delay measurement data for statistical analysis.

## 4.2 Kernel Density Estimation

Once the profile of each landmark is built, the second step is the estimation of the joint distribution of $(G_i, D_i)$, where $G_i$ represents the great circle distance between active landmark $L_i$, $i \in \mathcal{L}_a$ and the target $\tau$, and $D_i$ is the measured delay between $L_i$ and $\tau$. The joint probability density function of $(G_i, D_i)$ is denoted by $f_{G_i,D_i}(g, d)$. The sample data to be collected is represented as follows:

$$\mathcal{S}_i = \left\{ (g_{ij}, d_{ij}^{(l)}) : j \in \mathcal{L}_a, 1 \leq l \leq m \right\}, \quad i \in \mathcal{L}_a, \tag{4.1}$$

where $m$ is the number of delay measurements taken between a given pair of landmarks $L_i$ and $L_j$. In our experiments, $m = 5 \times 4 \times 24 \times 7$, since 5 delay measurements from landmark $L_i$ to landmark $L_j$, $j \neq i$, were taken once every 15 minutes over a period of one week. Let $M \triangleq |\mathcal{S}_i|$ denote the total number of delay measurements taken from a given landmark. For a set of 85 active landmarks, we have $M = 84m = 282,240$.

We apply the following kernel density estimator [23–25]:

$$\hat{f}_{i,\mathbf{H}}(g,d) = \frac{1}{M \det(\mathbf{H})} \sum_{j \in \mathcal{L}_a \setminus \{i\}} \sum_{l=1}^{m} \mathcal{K}((g - g_{ij}, d - d_{ij}^{(l)})\mathbf{H}^{-1}),$$

where $(g, d)$ is a vector consisting of great circle distance $g$ and delay $d$; $\mathbf{H}$ is a nonsingular matrix, called the *bandwidth matrix*; and $\mathcal{K}$ denotes the kernel. In our experimental work, we use a diagonal bandwidth matrix and a Gaussian kernel:

$$\mathbf{H} = \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix}, \ \mathcal{K}(g,d) = \frac{1}{2\pi} e^{-\frac{1}{2}(g^2+d^2)}. \tag{4.2}$$

Thus, the kernel density estimator becomes

$$\hat{f}_{i,\mathbf{H}}(g,d) = \frac{1}{2\pi h_1 h_2} \sum_{j \in \mathcal{L} \setminus \{i\}} \sum_{l=1}^{m} e^{-\frac{1}{2}\left[\left(\frac{g-g_{ij}}{h_1}\right)^2 + \left(\frac{d-d_{ij}^{(l)}}{h_2}\right)^2\right]}. \tag{4.3}$$

Several methods are available for choosing the bandwidth parameters $h_1$ and $h_2$. Popular choices include various rules-of-thumb, bootstrap methods, plug-in methods, unbiased cross validation, and biased cross validation. Scott's rule-of-thumb is given by [23, 24]

$$\hat{h}_j = M^{-1/6} \hat{\sigma}_j, \quad j \in \{1, 2\}. \tag{4.4}$$

Although Scott's rule-of-thumb choice of bandwidth parameters makes normality assumptions of underlying unknown distribution, we prefer this method due to its low complexity, i.e., $O(M)$ as opposed to $O(M^2)$ for the other methods. This is especially important as we deal with large data sets (e.g., on the order of 250,000 samples).

## 4.3 Application of Force-Directed Method

We employ a force-directed algorithm as an approximation algorithm to maximize the likelihood of the target location estimate given the delay measurement data. The force-directed method iteratively applies a force on the target proportional to the gradient of the estimated conditional distribution of distance from each landmark to the target given the delay. At each step of the algorithm, the resultant of the forces from all landmarks is calculated and then the target location estimate is moved in accordance with the resultant force. Thus, our algorithm combines the force-directed method with gradient ascent optimization. The initial estimate of the target location can be set as the landmark with the shortest delay to the target.

The gradient ascent steps $\{\eta_i\}$ form a decreasing sequence converging to zero, to ensure the convergence of the force-directed method. The initial gradient ascent step $\eta_0$ is chosen to be such that the target is moved a given distance from its initial position (e.g., 100 km, which is the magnitude of $10^8$ for the rule-of-thumb bandwidth). The algorithm stops when the target moved less than a value $\epsilon$, where $\epsilon$ is chosen in such a way to achieve a tradeoff between computational overhead and accuracy requirement.

Since the landmarks and targets are located on the earth, great circle distances must be considered. We use the WGS-84 ellipsoid [26] as a model for Earth and apply the Vincenty formulas to compute great circle distances [27]. We have implemented the direct and inverse Vincenty's formula in two functions.

*Direct Vincenty Formula:*

$$((\varphi_2, \lambda_2), b_2) = v_{\text{fwd}}((\varphi_1, \lambda_1), b_1, g), \tag{4.5}$$

which calculates the destination point $(\varphi_2, \lambda_2)$ and the final bearing $b_2$ given the starting point $(\varphi_1, \lambda_1)$, initial bearing $b_1$, and the great circle distance $g$ from the starting point to the destination.

*Inverse Vincenty Formula:*

$$(g, b_1, b_2) = v_{\text{inv}}((\varphi_1, \lambda_1), (\varphi_2, \lambda_2)), \tag{4.6}$$

which calculates the great circle distance $g$, the initial bearing $b_1$, and the final bearing $b_2$ given the starting point $(\varphi_1, \lambda_1)$ and the destination point $(\varphi_2, \lambda_2)$

Our proposed force-directed steepest ascent algorithm is summarized in Algorithm 2.

**Algorithm 2** SG: Statistical Geolocation Algorithm

**F1.** Start with a guess of the latitude and longitude of the target $(\varphi_\tau^{(0)}, \lambda_\tau^{(0)})$. Initialize $k \leftarrow 0$.

**F2.** Calculate the distance and final bearing from each landmark to the target using the inverse Vincenty formula:

$$(g_i^{(k)}, b_i^{(k)}) \leftarrow v_{\text{inv}}((\varphi_i,\, \lambda_i), (\varphi_\tau^{(k)},\, \lambda_\tau^{(k)})), \quad i \in \mathcal{L}_a.$$

**F3.** Execute one step of gradient ascent:

$$l_i^{(k)} \leftarrow g_i^{(k)} + \eta_k \hat{f}'_{G_i|D_i}(g_i^{(k)} \mid d_{i\tau}), \quad i \in \mathcal{L}_a.$$

**F4.** For each $i \in \mathcal{L}_a$ calculate the force vector $\mathbf{F}_i^{(k)}$ as follows:

**If** $\hat{f}_{G_i|D_i}(l_i^{(k)} \mid d_{i\tau}) > \hat{f}_{G_i|D_i}(g_i^{(k)} \mid d_{i\tau})$ **then**

$$|\mathbf{F}_i^{(k)}| \leftarrow l_i^{(k)} - g_i^{(k)};\ \text{bear}(\mathbf{F}_i^{(k)}) \leftarrow b_i^{(k)}$$

**Else** $\mathbf{F}_i^{(k)} \leftarrow \mathbf{0}$.

**F5.** Calculate the resultant force vector

$$\mathbf{F} = |\mathcal{L}_a| \text{gm}(\mathbf{F}_i^{(k)} : i \in \mathcal{L}_a)$$

**F6.** Move the target location estimate in the direction of the resultant force using the direct Vincenty formula:

$$(\varphi_\tau^{(k+1)},\, \lambda_\tau^{(k+1)}) \leftarrow v_{\text{fwd}}((\varphi_\tau^{(k)}, \lambda_\tau^{(k)}), \text{bear}(\mathbf{F}), |\mathbf{F}|)$$

Increment $k$ by one.
**If** target estimate moved more than $\varepsilon$ **then** go to **F2**.
**Else STOP**

The conditional pdf estimate $\hat{f}_{G_i|D_i}(g|d)$ in **F3** and **F4** can easily be obtained from the joint kernel density estimate (4.3). In step **F4**, a force vector $\mathbf{F}$, in geographical coordinates, is represented as being comprised of a magnitude $|\mathbf{F}|$ and a bearing bear($\mathbf{F}$). This is similar

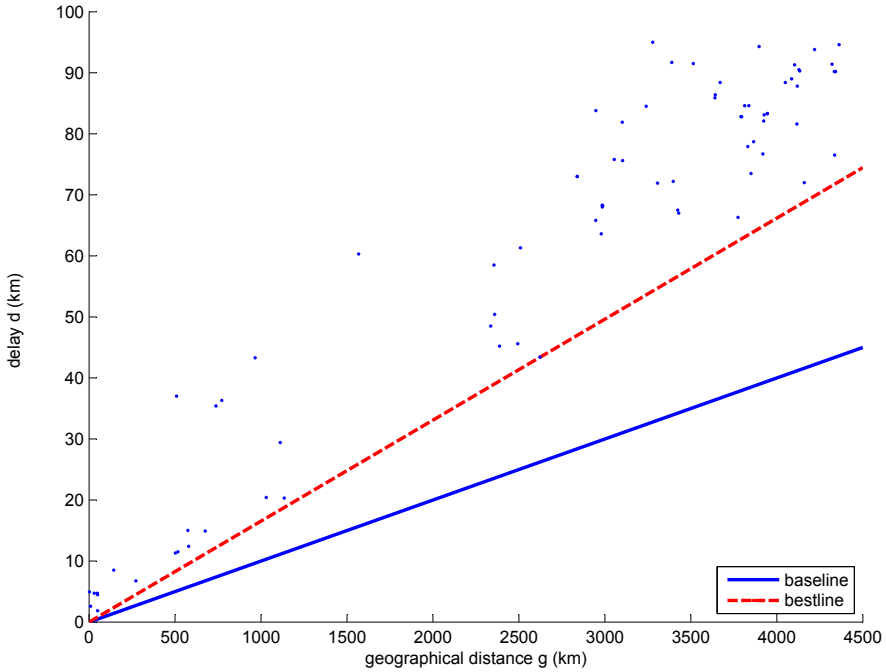


Figure 4.2: Scatterplot of distance and delay from *planet1.cs.stanford.edu* to 79 other PlanetLab nodes across the U.S. (see Fig. 4.1).

to the magnitude-phase representation in the complex plane. Referring to step **F5**, the operator gm(·) (geographical mean) computes the centroid of a set of points on a spherical surface. To compute the geographical mean, we make use of the built-in Matlab function MEANM. When the algorithm terminates in Step **F6**, the estimated location of the target is given by $(\varphi_\tau^{(k)}, \lambda_\tau^{(k)})$. The initial target location in Step **F1** can be obtained by applying a computationally simple geolocation method such as SPing or GeoPing [2,6].

## 4.4 Experimental Results

We conducted experiments over the PlanetLab network using 85 landmarks. The distribution of the landmarks over the continental U.S. is illustrated in Fig. 4.1. The PlanetLab database includes information on the latitude and longitude of each of the PlanetLab nodes. We used the CoMon project of PlanetLab to retrieve a list of the active nodes, filtered to

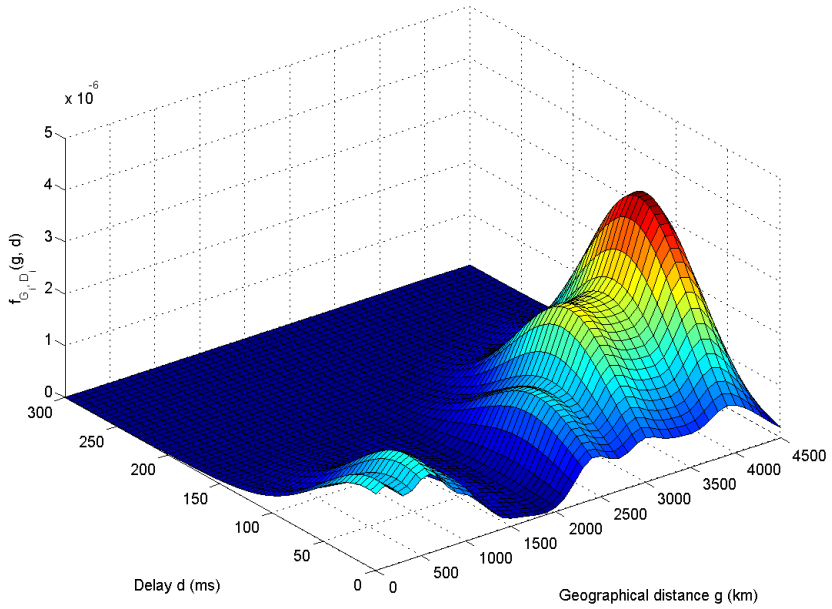


Figure 4.3: Kernel density estimate of bivariate distribution of distance and delay using Gaussian kernel for *planet1.cs.stanford.edu*.

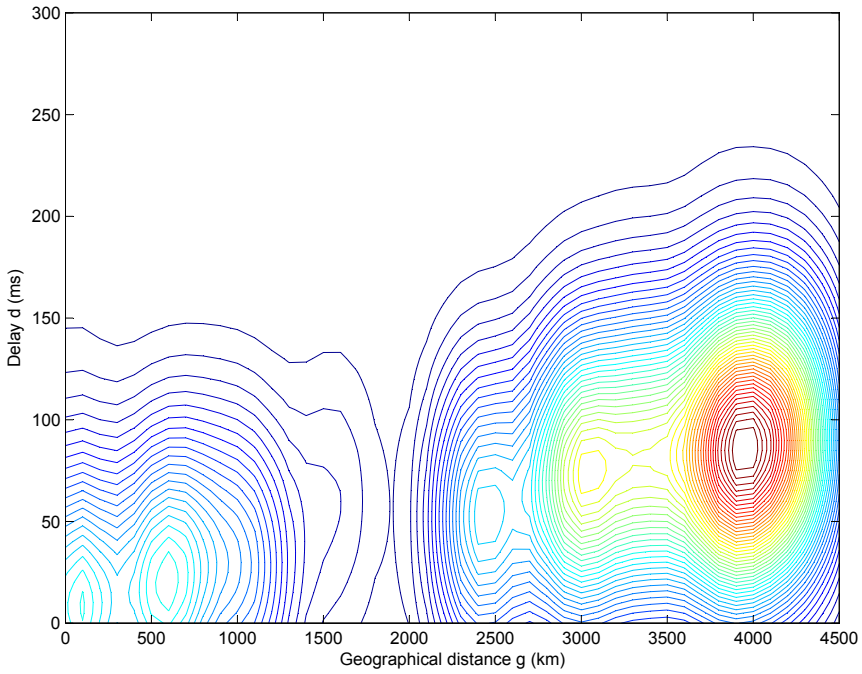


Figure 4.4: Contour plot of kernel density estimate for *planet1.cs.stanford.edu*.

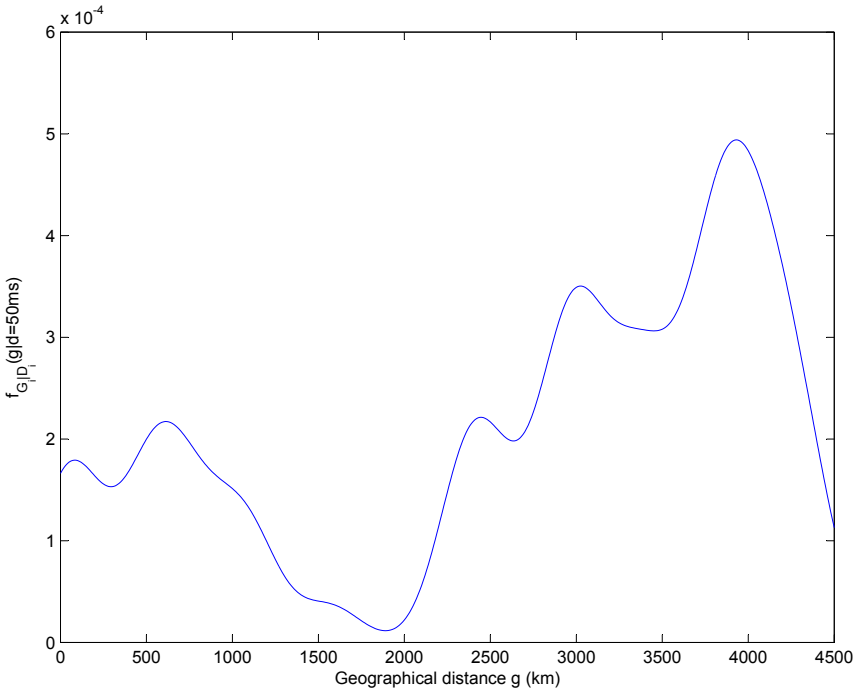


Figure 4.5: Estimated conditional pdf of distance from *planet1.cs.stanford.edu* to a target, given a delay of 50 ms.

select only one node per site. By means of a geocoding webpage written using JavaScript and the Google Map API, we filtered out a total of 93 sites located in the continental U.S. We tested each of these sites, of which only 85 nodes responded to *ping* commands (the others had firewall constraints).

We uploaded and executed a Python script in a distributed manner using the *codeploy* tool and saved the output in a log file. The log files were later downloaded and parsed using another Python script, and the measurement results were placed in comma separated value (CSV) files. As a result of our delay measurements over PlanetLab, we obtained 85 scatterplots and kernel density estimates of the joint pdf of distance and delay from each landmark to the target. Fig. 4.3 shows the KDE surface obtained at the PlanetLab node *planet1.cs.stanford.edu*. A contour plot of the kernel density estimate for the same landmark node is illustrated in Fig. 4.4. For this landmark, the conditional density of geographical distance given a 50 ms delay is shown in Fig. 4.5

The kernel density estimates were applied to the force-directed algorithm described in Section 4.3 to obtain the estimate of the target location. We validated the proposed geolocation scheme by removing each landmark from the set of all landmarks, and running our algorithms with the removed landmark as the target and the remaining landmarks. The initial target location estimate is the landmark which is closest from the point of view of RTT delay. The force-directed algorithm is designed to iteratively push the initial location estimate towards its true location, based on conditional distributions of geographical distance given delay. We observed that when the initial estimate is far from the real position of the target, our algorithm improves the estimate dramatically. However, when the initial estimate is close to the target, not much improvement is observed. To improve the resolution and accuracy, one has to increase the number of landmarks.

For comparison, we have implemented and executed the CBG and SPing algorithms. The CBG algorithm failed three times, yielding an empty confidence region. We removed these cases from the CBG statistics. In other five cases the confidence region did not include the target. By reducing the large errors, the average error of our statistical approach is

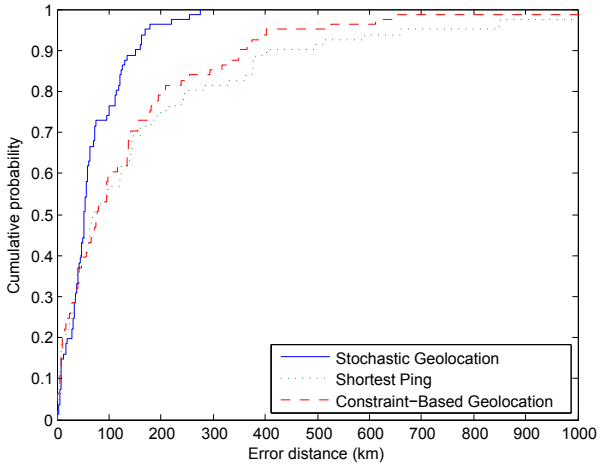


Figure 4.6: Cumulative distribution function of estimation error: statistical geolocation (SG), CBG, and SPing.

92 km. This is a dramatic improvement compared to 141 km for CBG and 184 km for SPing. The median error also decreases to 53 km, in comparison to 73 km for SPing and 78 km for CBG. We note that 77% of the location estimates from SG estimates were in the 100 km range, compared to 59% for CBG and 57% for SPing. Furthermore, 15% of the CBG estimates and 19% of the SPing estimations had an error of 300 km or more, while all but one of the SG estimates falled within the 300 km mark. Fig. 4.6 shows plots of the cumulative distribution function (cdf) of the estimation error for SPing, CBG, and SG. From this figure, it is clear that the statistical geolocation scheme is significantly more accurate than the CBG and SPing.

Table 4.1 displays the geolocation error performance of SPing, CBG, and SG. The error statistics shown in the left-hand column are the mean error, median error, maximum error, standard deviation of error, first quartile, and third quartile. All of the error values shown in the table are in units of km. In terms of mean error, SG shows a significant improvement over CBG, which in turn shows a significant improvement over SPing. The median errors of SPing and CBG are similar, while SG has a markedly smaller median error. Similarly, whereas the maximum error values for SPing and CBG are approximately the same, that of SG is about a factor of two smaller. Interestingly, the standard deviation of the error is

| Error [km] | SPing | CBG | SG |
|---|---|---|---|
| mean | 184 | 141 | 92 |
| median | 73 | 78 | 53 |
| maximum | 2167 | 2155 | 1054 |
| std. dev. | 309 | 176 | 238 |
| 1st quartile | 30 | 28 | 32 |
| 3rd quartile | 198 | 180 | 99 |

Table 4.1: Accuracy comparison of SPing, CBG, and SG.

smaller for CBG than for SG. The first quartile of the errors are approximately the same for all three schemes, but SG clearly outperforms the other two schemes in terms of the third quartile of error. In summary, the SG scheme appears to provide significantly higher accuracy than SG and CBG. There is however, room for improvement, as indicated by the result for the standard deviation of error. Aspects of the SG scheme that could be refined further include the kernel density estimation approach and the force-directed gradient ascent algorithm.

## 4.5 Analysis and Interpretation

We proposed a statistical approach to geolocation of Internet hosts, based on a the collection of delay measurements among a set of landmark nodes. In contrast to earlier measurement-based geolocation schemes, which provide loose deterministic bounds on the target location, the proposed scheme captures the statistical variations in Internet delay measurements. Besides the collection of active delay measurements, the key elements of the approach include kernel density estimation to obtain an estimate of the joint density function of the geographical distances and delays between landmarks, and a force-directed algorithm to move the target location estimate towards a point that maximizes the likelihood function.

We conducted experiments over PlanetLab using 85 landmark nodes. Our results show a significant improvement in accuracy over the previous approaches, in particular, CBG and SPing.

# Chapter 5: Conclusions

This thesis brings two major contributions in the area of geolocation of Internet hosts. In order to test thoroughly these advancements, we have created a measurement framework using PlanetLab infrastructure. We have collected our test data by performing measurements amongst the 81-85 landmarks with unique location provided by Planetlab.

The first contribution in the class of purely-delay based algorithms creates a mathematical framework which generalizes well-known geolocation schemes like GeoPing and Shortest Ping, by expressing the objective function as a combination of a proximity measure and a norm. Using a careful choice of the proximity measure and norm, one can discern between informative landmarks (whose measurements provide information about the target location), and non-informative landmark (which do not provide useful information with regard to target location). Using a real-data study, we find two classes of measures which provide superior robustness and accuracy in comparison to the GeoPing and Shortest Ping methods. The first class is provided by combining the $k^{th}$ order normalized delay with the Manhattan norm, which gives the Canberra distance (this is known to be a measure which is biased around origin and very sensitive for values close to zero). We found that other $p$-norms with $1/2 < p < 1$ give also similar good results. The second class is given by using the variance of measurements of a target as a measure of its reliability. This class uses Mahalanobis/Normalized Euclidean norm in combination with the $k^{th}$ order (unnormalized) delay.

The second major contribution of this thesis, in the area of algorithms which are based on both delay and location of the landmarks, proposes for the first time a statistical approach to the geolocation of Internet hosts. Unlike previous algorithms which provided only loose deterministic bounds on the target location, this method captures statistical variation

of the active delay measurements by constructing a profile for each landmark using kernel density estimation of the joint probability density function of delay/distance measurements. This estimate is later used by a force-directed algorithm to maximize the likelihood function. Using empirical data, the output estimate of this force-based method is proven to reduce the geolocation error, by having much improved accuracy for the targets which other algorithms either fail to locate completely, or give large errors. We have used the collected measurements to run our algorithms and demonstrate the superior performance of the proposed geolocation schemes in comparison to the existing algoritms.

# Appendix A: Vincenty's Direct and Inverse Formulae

Vincenty's Formulae are two algorithms used in geodesy to accurately calculate the great circle (ellipsoidal) distance between two points. The model used is the WGS-84 standard, which approximates the Earth surface with an ellipsoid. Using rapidly converging series for the distance and angle over the surface of the ellipsoid, the formulae are accurate within 0.5 mm (the actual distance might be slightly different due to the differences in elevation). The MATLAB implementation of these formulae follows [27] and its Javascript implementation in [28].

The direct Vincenty formula takes as input the initial latitude, longitude, bearing (compass bearing at the starting point), and the travel distance. It returns the latitude and longitude of the destination point, and the final bearing (compass bearing at the destination point). The MATLAB code is listed below:

```
function [lat2, lon2, revAz] = gcdest(lat1, lon1, brng, dist)
%GCDEST     Destination point, given start point, distance and bearing
%   [LAT2, LON2, REVAZ] = GCDEST(LAT1, LON1, BRNG, DIST) calculates the
%   destination point P2(LAT2,LON2) and the final bearing REVAZ, given the
%   start point P1(LAT1,LON1), bearing BRNG and the great circle distance
%   DIST between P1 and P2

% WGS-84 ellipsoid
% major, minor, and flattening of the Earth
a = 6378.137; % km (+/-2m)
b = 6356.7523142; % km
f = 1/298.257223563;

s = dist;
alpha1 = brng * pi/180;
sinAlpha1 = sin(alpha1);
cosAlpha1 = cos(alpha1);
```

```
18
19 tanU1 = (1-f) * tan(lat1 * pi/180);
20 cosU1 = 1 / sqrt((1 + tanU1*tanU1));
21 sinU1 = tanU1*cosU1;
22 sigma1 = atan2(tanU1, cosAlpha1);
23 sinAlpha = cosU1 * sinAlpha1;
24 cosSqAlpha = 1 - sinAlpha*sinAlpha;
25 uSq = cosSqAlpha * (a*a - b*b) / (b*b);
26 A = 1 + uSq/16384*(4096+uSq*(-768+uSq*(320-175*uSq)));
27 B = uSq/1024 * (256+uSq*(-128+uSq*(74-47*uSq)));
28
29 sigma = s / (b*A);
30 sigmaP = 2*pi;
31 while abs(sigma-sigmaP) > 1e-12
32     cos2SigmaM = cos(2*sigma1 + sigma);
33     sinSigma = sin(sigma); cosSigma = cos(sigma);
34     deltaSigma = B*sinSigma*(cos2SigmaM+B/4* ...
35         (cosSigma*(-1+2*cos2SigmaM*cos2SigmaM)- ...
36         B/6*cos2SigmaM*(-3+4*sinSigma*sinSigma)* ...
37         (-3+4*cos2SigmaM*cos2SigmaM)));
38     sigmaP = sigma;
39     sigma = s / (b*A) + deltaSigma;
40 end
41
42 tmp = sinU1*sinSigma - cosU1*cosSigma*cosAlpha1;
43 lat2 = atan2(sinU1*cosSigma + cosU1*sinSigma*cosAlpha1, ...
44     (1-f)*sqrt(sinAlpha*sinAlpha + tmp*tmp));
45 lambda = atan2(sinSigma*sinAlpha1, ...
46     cosU1*cosSigma - sinU1*sinSigma*cosAlpha1);
47 C = f/16*cosSqAlpha*(4+f*(4-3*cosSqAlpha));
48 L = lambda - (1-C) * f * sinAlpha * ...
49     (sigma + C*sinSigma*(cos2SigmaM+C*cosSigma* ...
50     (-1+2*cos2SigmaM*cos2SigmaM)));
51
52 lat2 = lat2 * 180/pi;
```

```
53 lon2 = lon1 + L * 180/pi;
54 if nargout > 2
55     revAz = atan2(sinAlpha, -tmp);  % final bearing
56     revAz = mod(revAz,2*pi) * 180/pi;
57 end
```

The inverse Vincenty formula takes as input the latitude and longitude of the initial and destination points. It returns the great circle (ellipsoidal) distance between the two points and the initial and final compass bearings. The MATLAB code is listed below:

```
1  function [s, alpha1, alpha2] = gcdist(lat1, lon1, lat2, lon2)
2  %GCDIST     Great circle distance using Vincenty's formula
3  %   [S, ALPHA1, ALPHA2] = GCDIST(LAT1, LON1, LAT2, LON2) calculates the
4  %   great circle distance S between P1 and P2 using Vincenty's formula,
5  %   where P1(LAT1,LON1) is the start point, P2(LAT2,LON2) is the
6  %   destination point, ALPHA1 is the initial bearing, and ALPHA2 is the
7  %   final bearing in the P1->P2 direction
8
9  % WGS-84 ellipsoid
10 % major, minor, and flattening of the Earth
11 a = 6378.137; % km (+/-2m)
12 b = 6356.7523142; % km
13 f = 1/298.257223563;
14
15 L = (lon2-lon1) * pi/180;
16 U1 = atan((1-f) * tan(lat1 * pi/180));
17 U2 = atan((1-f) * tan(lat2 * pi/180));
18 sinU1 = sin(U1); cosU1 = cos(U1);
19 sinU2 = sin(U2); cosU2 = cos(U2);
20
21 lambda = L; lambdaP = inf; iterLimit = 20;
22 while abs(lambda - lambdaP) > 1e-12 && iterLimit > 0
```

```
23      sinLambda = sin(lambda); cosLambda = cos(lambda);
24      sinSigma = sqrt((cosU2*sinLambda)^2 + ...
25          (cosU1*sinU2-sinU1*cosU2*cosLambda)^2);
26      if sinSigma == 0
27          s = 0;
28          return;
29      end % if
30      cosSigma = sinU1*sinU2 + cosU1*cosU2*cosLambda;
31      sigma = atan2(sinSigma, cosSigma);
32      sinAlpha = cosU1 * cosU2 * sinLambda / sinSigma;
33      cosSqAlpha = 1 - sinAlpha^2;
34      cos2SigmaM = cosSigma - 2*sinU1*sinU2/cosSqAlpha;
35      if isnan(cos2SigmaM)
36          cos2SigmaM = 0; % equatorial line: cosSqAlpha=0
37      end %if
38      C = f/16*cosSqAlpha*(4+f*(4-3*cosSqAlpha));
39      lambdaP = lambda;
40      lambda = L + (1-C) * f * sinAlpha * ...
41        (sigma + C*sinSigma*(cos2SigmaM+ ...
42        C*cosSigma*(-1+2*cos2SigmaM*cos2SigmaM)));
43      iterLimit = iterLimit - 1;
44  end %while
45
46  if iterLimit==0 % formula failed to converge
47      s = NaN;
48      return;
49  end %if
50
51  uSq = cosSqAlpha * (a*a - b*b) / (b*b);
52  A = 1 + uSq/16384*(4096+uSq*(-768+uSq*(320-175*uSq)));
53  B = uSq/1024 * (256+uSq*(-128+uSq*(74-47*uSq)));
54  deltaSigma = B*sinSigma* ...
55      (cos2SigmaM+B/4*(cosSigma*(-1+2*cos2SigmaM*cos2SigmaM)- ...
56      B/6*cos2SigmaM*(-3+4*sinSigma*sinSigma)* ...
57          (-3+4*cos2SigmaM*cos2SigmaM)));
```

```
58  s = b*A*(sigma-deltaSigma);
59  if nargout > 1
60      alpha1 = atan2(cos(U2)*sin(lambda), ...
61          cos(U1)*sin(U2)-sin(U1)*cos(U2)*cos(lambda));
62      alpha1 = mod(alpha1,2*pi) * 180/pi;
63  end
64  if nargout > 2
65      alpha2 = atan2(cos(U1)*sin(lambda), ...
66          -sin(U1)*cos(U2)+cos(U1)*sin(U2)*cos(lambda));
67      alpha2 = mod(alpha2,2*pi) * 180/pi;
68  end
```

# Appendix B: Best Line as Solution of a Linear Programming Problem in Constraint-Based Geolocation

As outlined in [4, 5] the packet propagation speed on the Internet is at most the speed of light through the optical fiber cable, which in turn is about 2/3 of the speed of light. This restriction induces circle-like bounds on the location of the target. In the delay-distance plane, this constraint translates in a line, which [4] calls *baseline*'. The equation of this line is $g = \frac{1}{100}d$, where $g$ is the geographical distance in kilometers and $d$ is the delay in milliseconds, as seen in Fig. B.1. However, the bounds on geographical distance given by the baseline are too loose to be of practical interest. To obtain tighter bounds, Gueye et. al. in [4, 5] construct a "best" linear lower bound based on inter-landmark delay and distance measurements, which they call *bestline*. If we denote by $d_i$ and $g_i$ the geographical distance in kilometers, respectively the 2.5 percentile delay in milliseconds from current landmark to landmark $i$, with $1 \leq i \leq n$, then the slope $m$ and the intercept $b$ of the bestline can be expressed as the solution of the following linear programming problem:

$$\text{minimize} \quad \sum_{i=1}^{n}(d_i - mg_i - b) \tag{B.1}$$

$$\text{subject to} \quad d_i - mg_i - b \geq 0 \quad , 1 \leq i \leq n \tag{B.2}$$

$$m \geq \frac{1}{100} \tag{B.3}$$

$$b \geq 0 \tag{B.4}$$

As seen in Fig. B.1, the objective function B.1 to be minimized is the sum of the $y$-distances between the distance-delay points and the bestline. The set of constraints in B.2 simply expresses that each point of coordinate $(g_i, d_i)$ is situated above the bestline in the Fig. B.1 graph. The constraint B.3 is the mathematical formulation that the slope of the bestline should be at least as large as the slope of the baseline, to guarantee there is no interesection

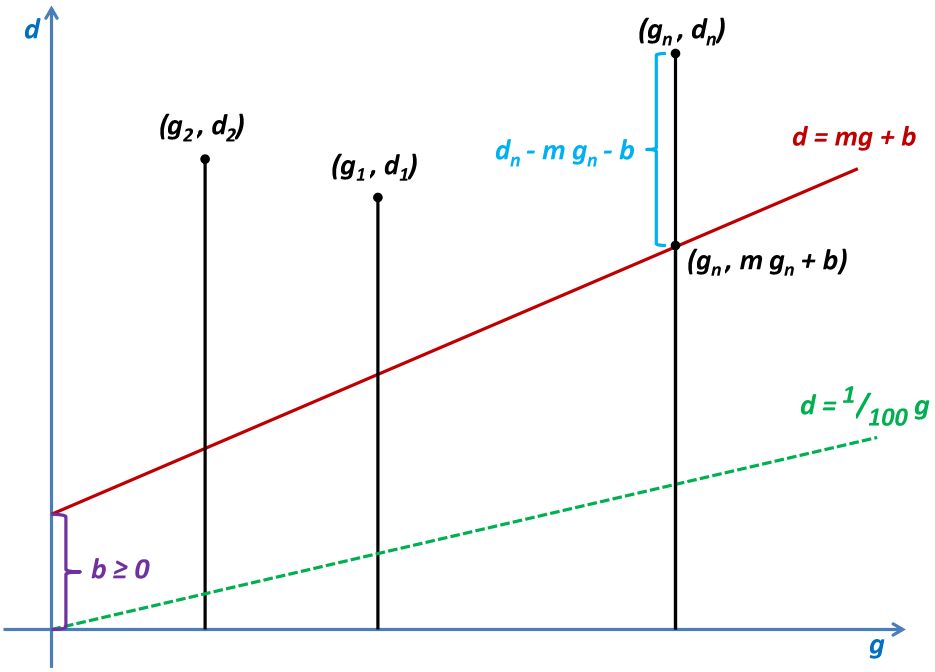


Figure B.1: Construction of "bestline" for Constraint-Based Geolocation algorithm.

between baseline and bestline for positive distances $g$. Thus, the bestline should be situated entirely above the baseline for positive $g$. The final constraint B.4 prevents the negative is also a necessary condition for the bestline to be situated above baseline for all positive distances $g$.

We can also notice one of the drawbacks of the Constraint-Based Geolocation scheme: just because all the measured inter-landmark distance-delay pairs falled above the bestline does not guarantee that the distance-delay measurement pair for the target is also situated above the bestline. If the measurement pair falls below the bestline, this geolocation scheme is guaranteed to provide an estimated region of the target which does not contain the actual location of the target.

The objective function can be written as:

$$\sum_{i=1}^{n}(d_i - mg_i - b) = \sum_{i=1}^{n} d_i - \left(\sum_{i=1}^{n} g_i\right) m - nb \tag{B.5}$$

Since the first term on the right hand side does not depend on $m$ or $b$, it suffices to minimize the simplified objective function: $-\left(\sum_{i=1}^{n} g_i\right) m - nb$. Thus, the linear programming problem can be written in the following form:

$$\text{minimize} \quad -\left(\sum_{i=1}^{n} g_i\right) m - nb \tag{B.6}$$

$$\text{subject to} \quad mg_i + b \leq d_i \qquad , 1 \leq i \leq n \tag{B.7}$$

$$-m \leq -\frac{1}{100} \tag{B.8}$$

$$-b \leq 0 \tag{B.9}$$

which becomes the matrix form:

$$\text{minimize} \quad c^T x \tag{B.10}$$

$$\text{subject to} \quad Ax \leq v \tag{B.11}$$

where:

$$c = \begin{bmatrix} -\sum_{i=1}^{n} g_i \\ -n \end{bmatrix}, \quad A = \begin{bmatrix} g_1 & b \\ g_2 & b \\ \vdots & \vdots \\ g_n & b \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad v = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ -\frac{1}{100} \\ 0 \end{bmatrix} \tag{B.12}$$

This is the form which is accepted by MATLAB's *linprog* function, which solves this linear programming problem returning the slope $m$ and intercept $b$ of the bestline.

# Appendix C: Notes on Kernel Density Estimation

This appendix provides details on the univariate and multivariate nonparametric density estimation, as well as three well-known bandwidth selection rules (Scott's rule of thumb, Silverman's rule of thumb and unbiased cross-validation). It follows closely classical texts [23–25, 29–31].

## C.1 Parametric Estimation

Suppose $X_1, X_2 \dots X_n$ are independent and identically distributed (i.i.d.) random variables with common probability density function $f(x;\theta)$, where $f$ is known and $\theta$ is unknown.

We would like to estimate $\theta$ with an estimator (random variable) $\hat{\theta} = \hat{\theta}(X_1, X_2 \dots X_n)$. We observe the values $X_1 = x_1,\ X_2 = x_2\ \dots X_n = x_n$ and estimate $\hat{\theta} = \hat{\theta}(x_1 \dots x_n)$. In order to achieve a good estimation, we need a criteria to decide which estimator performs best. The most common indicator of an estimator's performance is the Square Error (SE):

$$SE(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^2 \tag{C.1}$$

However, the square error is a random variable itself. Therefore, we need to minimize the expected value of the Square error, called Mean Square Error (MSE). The Mean Square Error reflects the average difference between the estimator and the underlying parameter, and is defined as follows:

$$\begin{aligned} MSE(\hat{\theta}, \theta) &= \mathrm{E}[(\hat{\theta} - \theta)^2] \\ &= \mathrm{E}[(\hat{\theta}(X_1 \dots X_n) - \theta)^2] \\ &= \int \cdots \int (\hat{\theta}(x_1, x_2 \dots x_n) - \theta)^2 f_{X_1 \dots X_n}(x_1, x_2 \dots x_n)\, dx_1 dx_2 \dots dx_n \end{aligned} \tag{C.2}$$

## C.2 Bias + Variance Theorem

The mean square error can be expressed as the sum between the variance of the estimator and its squared bias:

$$MSE(\hat{\theta}, \theta) = \mathrm{Var}(\hat{\theta}) + (bias(\hat{\theta}, \theta))^2 \tag{C.3}$$

where $bias(\hat{\theta}, \theta) = \mathrm{E}[\hat{\theta}] - \theta$. Indeed:

$$\begin{aligned} MSE(\hat{\theta}, \theta) &= \mathrm{E}[(\hat{\theta} - \theta)^2] \\ &= \mathrm{E}[(\hat{\theta} - \mathrm{E}[\hat{\theta}] + \mathrm{E}[\hat{\theta}] - \theta)^2] \\ &= \mathrm{E}[(\hat{\theta} - \mathrm{E}[\hat{\theta}])^2] + \mathrm{E}[2(\hat{\theta} - \mathrm{E}[\hat{\theta}])(\mathrm{E}[\hat{\theta}] - \theta)] + E[(\mathrm{E}[\hat{\theta}] - \theta)^2] \end{aligned}$$

The middle term of the right hand side cancels, and the last term inside the expectation operator is a constant, therefore:

$$\begin{aligned} MSE(\hat{\theta}, \theta) &= \mathrm{E}[(\hat{\theta} - \mathrm{E}[\hat{\theta}])^2] + (\mathrm{E}[\hat{\theta}] - \theta)^2 \\ &= \mathrm{Var}(\hat{\theta}) + (bias(\hat{\theta}, \theta)) \end{aligned}$$

## C.3 Nonparametric Density Estimation

Suppose $X_1, X_2 \ldots X_n$ are independent and identically distributed (i.i.d.) random variables with common probability density function $f(x)$, where $f$ is unknown. We estimate $f(x)$ by a random variable as $\hat{f}$ follows

$$f(x) = \hat{f}(x; X_1, X_2 \ldots X_n) \tag{C.4}$$

In terms of observed values, we observe $X_1 = x_1$, $X_2 = x_2 \ldots X_n = x_n$ and we estimate $f(x)$ as:

$$f(x) = \hat{f}(x; x_1, x_2 \ldots x_n) \tag{C.5}$$

As in Appendix C.1, we need a minimization criteria in order to find the best estimator. The most common used criteria is the Integrated Square Error (ISE) random variable:

$$\begin{aligned} ISE(\hat{f}, f) &= \int (\hat{f}(x) - f(x))^2 \, dx \\ &= \int (\hat{f}(x; X_1, X_2 \ldots, X_n) - f(x))^2 \, dx \end{aligned} \tag{C.6}$$

As ISE is a random variable, we minimize its expected value, called Mean Integrated Squared Error (MISE) in order to find the optimal estimator.

$$\begin{aligned} MISE(\hat{f}, f) &= \mathrm{E}[ISE(\hat{f}, f)] \\ &= \mathrm{E}\left[\int (\hat{f}(x; X_1, X_2 \ldots X_n)\right] \\ &= \int \cdots \int \left[\int (\hat{f}(x; x_1, x_2 \ldots x_n) - f(x))^2 f_{X_1 X_2 \ldots X_n}(x_1, x_2, \ldots x_n) \, dx\right] dx_1 \ldots dx_n \\ &= \int \left[\int \cdots \int (\hat{f}(x; x_1, x_2 \ldots x_n) - f(x))^2 f_{X_1 X_2 \ldots X_n}(x_1, x_2, \ldots x_n) \, dx_1 \ldots dx_n\right] dx \end{aligned} \tag{C.7}$$

## C.4 Univariate Kernel Density Estimation

Suppose $X_1, X_2 \ldots X_n$ are independent and identically distributed (i.i.d.) random variables with common probability density function $f(x)$, where $f$ is unknown. Then we can estimate

$f(x)$ by:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i) \tag{C.8}$$

where $K_h(x) = \frac{1}{h}K(\frac{x}{h})$. $K$ has the properties:

$$K(x) \geq 0 \tag{C.9}$$

$$K(x) = K(-x) \tag{C.10}$$

$$\int K(x)\,dx = 1 \tag{C.11}$$

If we observe the values $X_1 = x_1$, $X_2 = x_2 \ldots X_n = x_n$, then

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) \tag{C.12}$$

As a consequence, we have:

$$\int \hat{f}_h(x)\,dx = \frac{1}{n}\sum_{i=1}^{n} \int \frac{1}{h}K\left(\frac{x - x_i}{h}\right) dx = \frac{1}{n}n = 1 \tag{C.13}$$

Thus, $\hat{f}_h$ is a true probability density function.

## C.5 Bivariate Kernel Density Estimation (Diagonal Bandwidth)

Suppose we have $(X_1, Y_1)$, $(X_2, Y_2)$ $\ldots$ $(X_n, Y_n)$ independent and identically distributed bivariate random variables with common probability density function $f(x, y)$, which is unknown. Then we can estimate $f(x, y)$ by the folowing random variable (estimator):

$$\hat{f}_{h_1 h_2}(x, y) = \frac{1}{n h_1 h_2} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_1}, \frac{y - Y_i}{h_2}\right) = \frac{1}{n} \sum_{i=1}^{n} K_{h_1 h_2}(x - X_i, y - Y_i) \tag{C.14}$$

where K has the following properties:

$$K(x, y) \geq 0 \tag{C.15}$$

$$K(x, y) = K(-x, y) = K(x, -y) \tag{C.16}$$

$$\iint K(x, y)\, dxdy = 0 \tag{C.17}$$

If the observed values are $(X_1, Y_1)$, $(X_2, Y_2)$ $\ldots$ $(X_n, Y_n)$, then:

$$\hat{f}_{h_1 h_2}(x, y) = \frac{1}{n h_1 h_2} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h_1}, \frac{y - y_i}{h_2}\right) = \frac{1}{n} \sum_{i=1}^{n} K_{h_1 h_2}(x - x_i, y - y_i) \tag{C.18}$$

As a consequence, $\hat{f}_{h_1 h_2}$ is a true pdf:

$$\begin{aligned}\iint \hat{f}_{h_1 h_2}(x,y)\,dxdy &= \frac{1}{nh_1h_2}\sum_{i=1}^{n}\iint K\left(\frac{x-x_i}{h_1},\frac{y-y_i}{h_2}\right)dxdy \\ &= \frac{1}{n}\sum_{i=1}^{n}\iint \frac{1}{h_1h_2}K\left(\frac{x-x_i}{h_1},\frac{y-y_i}{h_2}\right)dxdy \\ &= \frac{1}{n}\sum_{i=1}^{n}\iint \frac{1}{h_1h_2}K(u,v)h_1h_2\,dxdy = \frac{1}{n}n = 1\end{aligned}$$

with the substitution $u = \frac{x-x_i}{h_1}$, $v = \frac{y-y_i}{h_2}$, and determinant of Jacobian matrix $h_1h_2$.

## C.6 General Multivariate Kernel Density Estimation

Suppose $\mathbf{X}_1,\ \mathbf{X}_2\ \ldots\ \mathbf{X}_n$ are independent and identically distributed random vectors with the common probability density function $f(\mathbf{x}) = f(x_1, x_2 \ldots x_d)$, where $f$ is unknown and $\mathbf{X}_i$ is a $d$-dimensional vector ($\mathbf{X}_i = [X_{i1}, X_{i2} \ldots X_{id}]^T$). Suppose that:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1d} \\ h_{21} & h_{22} & \cdots & h_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ h_{d1} & h_{d2} & \cdots & h_{dd} \end{bmatrix}$$

is a non-singular matrix with positive determinant ($\det(\mathbf{H}) > 0$). Then we can estimate the probability density function $f$ by the random variable $f_{\mathbf{H}}$ as follows:

$$\begin{aligned} f_{\mathbf{H}}(\mathbf{x}) &= \frac{1}{n\det(\mathbf{H})}\sum_{i=1}^{n}K(\mathbf{H}^{-1}(\mathbf{x}-\mathbf{X}_i)) \\ &= \frac{1}{n}\sum_{i=1}^{n}K_{\mathbf{H}}(\mathbf{H}^{-1}(\mathbf{x}-\mathbf{X}_i)) \end{aligned} \tag{C.19}$$

where $K_{\mathbf{H}}(\mathbf{x}) = \frac{1}{\det(\mathbf{H})} K(\mathbf{H}^{-1}x)$. $K$ should have the properties:

$$K(\mathbf{x}) \geq 0$$

$$K(x_1, x_2 \ldots - x_i \ldots x_n) = K(x_1, x_2 \ldots x_i \ldots x_n) \tag{C.20}$$

$$\int \cdots \int K(x_1, x_2 \ldots x_n)\, dx_1 \ldots dx_n = 1$$

Thus, if we observe the values $\mathbf{X}_1 = \mathbf{x}_1,\ \mathbf{X}_2 = \mathbf{x}_2\ \ldots\ \mathbf{X}_n = \mathbf{x}_n$ then the estimation of $f(\mathbf{x})$ is:

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n \det(\mathbf{H})} \sum_{i=1}^{n} K(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)) \tag{C.21}$$

By substituting $\mathbf{y} = \mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i)$ we can prove that the kernel integrates to 1:

$$\begin{aligned} \int \cdots \int \frac{1}{\det(\mathbf{H})} K(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i))\, d\mathbf{x} &= \int \cdots \int \frac{1}{\det(\mathbf{H})} K(\mathbf{y}) \det(\mathbf{H})\, d\mathbf{y} \\ &= \int \cdots \int K(\mathbf{y})\, d\mathbf{y} = 1 \end{aligned}$$

Therefore $f_{\mathbf{H}}$ is a true probability density function:

$$\begin{aligned} \int \cdots \int \hat{f}_{\mathbf{H}}(\mathbf{x})\, d\mathbf{x} &= \frac{1}{n} \sum_{i=1}^{n} \int \cdots \int \frac{1}{\det(\mathbf{H})} K(\mathbf{H}^{-1}(\mathbf{x} - \mathbf{x}_i))\, d\mathbf{x} \\ &= \frac{1}{n} n = 1 \end{aligned}$$

If $\mathbf{H}$ is a diagonal matrix:

$$\mathbf{H} = \text{diag}(h_1, h_2 \dots h_d) = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_d \end{bmatrix}$$

then we get the $n$-dimensional version of the bivariate diagonal bandwidth kernel density estimate:

$$\det(\mathbf{H}) = \begin{vmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_d \end{vmatrix} = h_1 h_2 \dots h_d$$

$$\mathbf{H}^{-1} = \text{diag}(\frac{1}{h_1}, \frac{1}{h_2} \cdots \frac{1}{h_d}) = \begin{bmatrix} 1/h_1 & 0 & \cdots & 0 \\ 0 & 1/h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/h_d \end{bmatrix}$$

and

$$\begin{aligned} \hat{f}_{\mathbf{H}}(\mathbf{x}) &= \hat{f}_{\mathbf{H}}(x_1, x_2 \dots x_d) \\ &= \frac{1}{n h_1 h_2 \dots h_d} \sum_{i=1}^{n} K\left( \frac{x_1 - x_{i1}}{h_1}, \frac{x_2 - x_{i2}}{h_2} \cdots \frac{x_d - x_{id}}{h_d} \right) \end{aligned} \tag{C.22}$$

## C.7 Product and Radial Multivariate Kernels

The most popular multivariate kernels constructed from univariate kernels are the product and radial multivariate kernels. The product or multiplicative kernel is derived from

multiplication of $n$ (possibly different) univariate kernels:

$$K(\mathbf{x}) = K(x_1, x_2 \dots x_d) = K(x_1)K(x_2) \dots K(x_d) \tag{C.23}$$

Another way of deriving a multivariate kernel from a univariate one is to consider rotation invariant multivariate kernels:

$$\begin{aligned} K(\mathbf{x}) &= \frac{1}{c} K\left(\|\mathbf{x}\|_2\right) \\ &= \frac{1}{c} K\left(\sqrt{\mathbf{x}^T\mathbf{x}}\right) \\ &= \frac{1}{c} K\left(\sqrt{x_1^2 + x_2^2 + \ldots + x_d^2}\right) \end{aligned} \tag{C.24}$$

where $c$ is chosen so that the kernel integrates to 1. Thus, the Gaussian kernel:

$$K(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} e^{-\frac{\|\mathbf{x}\|_2^2}{2}} \tag{C.25}$$

is a kernel which is both product and radial. This property makes it suitable as the choice kernel for our algorithms.

## C.8 Gaussian Kernel Density Estimation of Bivariate Probability Density Functions (Diagonal Bandwidth)

As mentioned in the previous section, the Gaussian kernel is a special kernel for which the product and the radial kernels are identical:

$$K(x, y) = K(x)K(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} \tag{C.26}$$

By plugging K and $\mathbf{H} = \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix}$ into C.22 we get:

$$\hat{f}_{h_1 h_2}(x, y) = \frac{1}{2\pi n h_1 h_2} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left[ \left( \frac{x - x_i}{h_1} \right)^2 + \left( \frac{y - y_i}{h_2} \right)^2 \right] \right\} \tag{C.27}$$

From C.27 we can get the conditional pdf and its derivative. The marginal pdf with respect to the second variable is:

$$\begin{aligned} \hat{f}_{h_2}(y) &= \int \hat{f}_{h_1 h_2}(x, y)\, dx \\ &= \int \frac{1}{2\pi n h_1 h_2} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left[ \left( \frac{x - x_i}{h_1} \right)^2 + \left( \frac{y - y_i}{h_2} \right)^2 \right] \right\} dx \\ &= \frac{1}{n h_2 \sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left( \frac{y - y_i}{h_2} \right)^2 \right\} \int \frac{1}{h_1 \sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left( \frac{x - x_i}{h_1} \right)^2 \right\} dx \\ &= \frac{1}{n h_2 \sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left( \frac{y - y_i}{h_2} \right)^2 \right\} dx \end{aligned} \tag{C.28}$$

Thus the conditional pdf is given by the following formula:

$$\begin{aligned} \hat{f}_{h_1 h_2}(x|y) &= \frac{\hat{f}_{h_1 h_2}(x, y)}{\hat{f}_{h_2}(y)} \\ &= \frac{\frac{1}{2\pi n h_1 h_2} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left[ \left( \frac{x - x_i}{h_1} \right)^2 + \left( \frac{y - y_i}{h_2} \right)^2 \right] \right\}}{\frac{1}{n h_2 \sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2} \left( \frac{y - y_i}{h_2} \right)^2 \right\} dx} \end{aligned} \tag{C.29}$$

## C.9 Rule-of-Thumb Bandwidth Selection

In order to use the kernel density estimation as part of an algorithm, it is important to have a procedure of automatic kernel bandwidth selection. One of the simplest methods

is the plug-in method proposed in [23]. The "rule of thumb" methods make normality assumption $N(\mu, \Sigma)$ of the underlying distribution. Under certain regularity conditions, one can approximate the bias and variance by using a second order (multivariate) Taylor expansion:

$$bias(\hat{f}_{\mathbf{H}}(\mathbf{x})), f(\mathbf{x}) \approx \frac{1}{4}\mu_2^2(K) \int [\mathrm{tr}(\mathbf{H}^T \mathcal{H}_f \mathbf{H})]^2 \, d\mathbf{x} \tag{C.30}$$

$$\mathrm{Var}(\hat{f}_{\mathbf{H}}(\mathbf{x}))) \approx \frac{1}{\det(\mathbf{H})} \|K\|_2^2 f(\mathbf{x}) \tag{C.31}$$

where $\mu_2^2(K)\mathbf{I}_d = \int \cdots \int \mathbf{x}\mathbf{x}^T K(\mathbf{x}) \; dx$, $\mathcal{H}_f$ is the Hessian matrix of second order partial derivatives of $f$, and $\|K\|_2^2$ is the $d$-dimensional squared $L_2$ norm of $K$. The Mean Integrated Square Error (MISE) can then be approximated by Asymptotic Mean Integrated Square Error (AMISE):

$$AMISE(\mathbf{H}) = \frac{1}{4}\mu_2^2(K) \int [\mathrm{tr}(\mathbf{H}^T \mathcal{H}_f \mathbf{H})]^2 \, d\mathbf{x} + \frac{1}{\det(\mathbf{H})} \|K\|_2^2 f(\mathbf{x}) \tag{C.32}$$

In the simplest case when both $f$ and $K$ are multivariate Gaussian random vectors distributed as $N(\mu, \Sigma)$ respectively $N(\mathbf{0}, \mathbf{I}_d)$, and both $\mathbf{H}$ and $\Sigma$ are diagonal matrices $\mathbf{H} = \mathrm{diag}(h_1, h_2, \ldots h_d)$ and $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2 \ldots \sigma_d^2)$, the optimal bandwidth which minimizes the $AMISE$ is derived in [31]:

$$h_j^\star = \left( \frac{4}{d+2} \right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}} \sigma_j \tag{C.33}$$

By replacing the theoretical standard deviation $\sigma_j$ with the sample standard deviation $\hat{\sigma}_j$, we get a generalization of Silverman's rule of thumb [23]:

$$h_j^\star = \left( \frac{4}{d+2} \right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}} \hat{\sigma}_j \tag{C.34}$$

As the first term $\left(\frac{4}{d+2}\right)^{1/(d+4)}$ is always between 0.924 and 1.059, one can ignore it, thus obtaining Scott's rule:

$$h_j^\star = n^{-\frac{1}{d+4}}\hat{\sigma_j} \tag{C.35}$$

However, in our case of interest ($d = 2$) Scott's rule and Silverman's rule are identical:

$$h_j^\star = n^{-\frac{1}{6}}\hat{\sigma_j} \tag{C.36}$$

## C.10 One-Dimensional Unbiased Cross-Validation

Another way of choosing the bandwidth is by using Unbiased Cross Validation (UCV) It starts with minimizing the Integrated Square Error (ISE):

$$\begin{aligned} ISE(\hat{f}_h, f) &= \int \left(\hat{f}_h(x) - f(x)\right)^2 dx \\ &= \int \left(\hat{f}_h(x)\right)^2 - 2\cdot\hat{f}_h(x)\cdot f(x) + (f(x))^2\, dx \\ &= \int \left(\hat{f}_h(x)\right)^2 dx - 2\cdot\int \hat{f}_h(x)\cdot f(x)\, dx + \int (f(x))^2\, dx \end{aligned} \tag{C.37}$$

Minimizing ISE is equivalent to minimizing:

$$\begin{aligned} CV(h) &= ISE(\hat{f}_h, f) - \int (f(x))^2\, dx \\ &= \underbrace{\int \left(\hat{f}_h(x)\right)^2 dx}_{\text{A}} - \underbrace{2\cdot\int \hat{f}_h(x)\cdot f(x) dx}_{\text{B}} \end{aligned} \tag{C.38}$$

Part A can be easily calculated by using the substitution $z = \frac{x-x_i}{h}$, which yields:

$$
\begin{aligned}
\int \left(\hat{f}_h(x)\right)^2 dx &= \int \left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)\right)^2 dx \\
&= \frac{1}{n^2h^2}\int \left(\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)\right)\left(\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right)\right) dx \\
&= \frac{1}{n^2h^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int K\left(\frac{x-x_i}{h}\right) K\left(\frac{x-x_j}{h}\right) dx \\
&= \frac{1}{n^2h^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int K(z) K\left(z-\frac{x_j-x_i}{h}\right) h\, dz \\
&= \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{j=1}^{n}\int K(z) K\left(\frac{x_j-x_i}{h}-z\right) dz \\
&= \frac{1}{n^2h}\sum_{i=1}^{n}\sum_{j=1}^{n} K * K\left(\frac{x_j-x_i}{h}\right)
\end{aligned}
\tag{C.39}
$$

To estimate part B we notice that if $X$ is distributed with the unknown probability density function $f(x)$ then:

$$
\mathrm{E}[\hat{f}_h(X)] = \int \hat{f}_h(x) f(x)\, dx \tag{C.40}
$$

Thus $\mathrm{E}[\hat{f}_h(X)]$ can be approximated by the average of the observed values:

$$
\mathrm{E}[\hat{f}_h(X)] \approx \frac{\hat{f}_h(x_1) + \hat{f}_h(x_2) + \cdots + \hat{f}_h(x_n)}{n} \approx \frac{1}{n}\sum_{i=1}^{n} \hat{f}_h(x_i) \tag{C.41}
$$

However, it is incorrect to use the $x_i$ observation to calculate $\hat{f}_h(x_i)$. Therefore we estimate

$\hat{f}_h(x_i)$ from the other $n-1$ observations:

$$\hat{f}_h(x) \approx \hat{f}_{h,-i}(x_i) \tag{C.42}$$

where $\hat{f}_{h,-i}(x_i)$ can be written using the kernel symmetry, $K(-x) = K(x)$, as:

$$\hat{f}_{h,-i}(x_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1}{h} K\left(\frac{x_i - x_j}{h}\right) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1}{h} K\left(\frac{x_j - x_i}{h}\right) \tag{C.43}$$

The final approximation is

$$\mathrm{E}[\hat{f}_h(X)] \approx \frac{1}{n(n-1)h} \sum_{i=1}^{n} \sum_{j \neq i} K\left(\frac{x_j - x_i}{h}\right) \tag{C.44}$$

By plugging A and B in the definition of $CV(h)$ we get:

$$CV(h) = \frac{1}{n^2 h} \sum_{i=1}^{n} \sum_{j \neq i} K * K\left(\frac{x_j - x_i}{h}\right) - \frac{2}{n(n-1)h} \sum_{i=1}^{n} \sum_{j \neq i} K\left(\frac{x_j - x_i}{h}\right) \tag{C.45}$$

The optimal $h$ is the one which minimizes $CV(h)$:

$$h^\star = \arg\min_h CV(h) \tag{C.46}$$

If $K$ is the Gaussian Kernel (i.e. pdf of a $N(0,1)$ random variable) then $K * K$ is also a Gaussian Kernel, pdf of a $N(0,2)$ random variable, with standard deviation: $\sigma = \sqrt{2}$.

## C.11 Two-Dimensional Unbiased Cross-Validation (Diagonal Bandwidth)

As in the one-dimensional case, the objective is to minimize the integrated square error:

$$\begin{aligned} ISE(\hat{f}_{h_1,h_2}, f) &= \iint \left(\hat{f}_{h_1,h_2}(x,y) - f(x,y)\right)^2 \, dxdy \\ &= \iint \left(\hat{f}_{h_1,h_2}(x,y)\right)^2 \, dxdy - 2\iint \hat{f}_{h_1,h_2}(x,y) f(x,y) \, dxdy + \iint (f(x,y))^2 \, dxdy \end{aligned} \tag{C.47}$$

The last term ($\iint (f(x,y))^2 \, dxdy$) does not depend on $h_1$ or $h_2$, therefore in order to minimize $ISE(\hat{f}_{h_1,h_2}, f)$ it is sufficient to minimize:

$$\begin{aligned} CV(h_1,h_2) &= ISE(\hat{f}_{h_1,h_2},\, f) - \iint (f(x,y))^2 \, dxdy \\ &= \iint \left(\hat{f}_{h_1,h_2}(x,y)\right)^2 \, dxdy - 2\iint \hat{f}_{h_1,h_2}(x,y) f(x,y) \, dxdy \\ &= \iint \left(\hat{f}_{h_1,h_2}(x,y)\right)^2 \, dxdy - 2\mathrm{E}[\hat{f}_{h_1,h_2}(X,Y)] \end{aligned} \tag{C.48}$$

where $(X,Y)$ is distributed with the probability density function $f(x,y)$

Using the substitution $u = \frac{x-x_i}{h_1}$, $v = \frac{y-y_i}{h_2}$ the first term of C.48 can be written as:

$$\begin{aligned}
\iint \left(\hat{f}_{h_1,h_2}(x,y)\right)^2 dxdy &= \iint \left(\frac{1}{nh_1h_2}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_1}, \frac{y-y_i}{h_2}\right)\right)^2 dxdy \\
&= \frac{1}{n^2{h_1}^2{h_2}^2} \iint \left(\sum_{i=1}^{n} K\left(\frac{x-x_i}{h_1}, \frac{y-y_i}{h_2}\right)\right)\left(\sum_{j=1}^{n} K\left(\frac{x-x_j}{h_1}, \frac{y-y_j}{h_2}\right)\right) dxdy \\
&= \frac{1}{n^2{h_1}^2{h_2}^2} \sum_{i=1}^{n}\sum_{j=1}^{n} \iint K\left(\frac{x-x_i}{h_1}, \frac{y-y_i}{h_2}\right) K\left(\frac{x-x_j}{h_1}, \frac{y-y_j}{h_2}\right) dxdy \\
&= \frac{1}{n^2{h_1}^2{h_2}^2} \sum_{i=1}^{n}\sum_{j=1}^{n} \iint K(u,v) K\left(u - \frac{x-x_j}{h_1}, v - \frac{y-y_j}{h_2}\right) h_1h_2\, dudv \\
&= \frac{1}{n^2{h_1}^2{h_2}^2} \sum_{i=1}^{n}\sum_{j=1}^{n} K * K\left(\frac{x_j-x_i}{h_1}, \frac{y_j-y_i}{h_2}\right)
\end{aligned} \tag{C.49}$$

As in the one-dimensional case, we approximate $\mathrm{E}[\hat{f}_{h_1,h_2}(X,Y)]$ with the sample average and use the symmetry property of the kernel $K(x,y) = K(-x,-y)$. In order to get an estimate of $\hat{f}_{h_1,h_2}(x_i,y_i)$ we skip the measurement with index $i$, so we use only $n-1$ measurements

to get the density estimate denoted by $\hat{f}_{h_1,h_2,-i}(x_i, y_i)$

$$\begin{aligned}
\mathrm{E}[\hat{f}_{h_1,h_2}(X,Y)] &\approx \frac{\hat{f}_{h_1,h_2}(x_1,y_1) + \hat{f}_{h_1,h_2}(x_2,y_2)\ldots\hat{f}_{h_1,h_2}(x_n,y_n)}{n} \\
&\approx \frac{1}{n}\sum_{i=1}^{n}\hat{f}_{h_1,h_2}(x_i,y_i) \\
&\approx \sum_{i=1}^{n}\hat{f}_{h_1,h_2,-i}(x_i,y_i) \\
&\approx \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{1}{h_1h_2}K\left(\frac{x_i-x_j}{h_1},\frac{y_i-y_j}{h_2}\right) \\
&\approx \sum_{i=1}^{n}\sum_{j\neq i}K\left(\frac{x_j-x_i}{h_1},\frac{y_j-y_i}{h_2}\right)
\end{aligned}$$

Therefore the objective function to be minimized is:

$$\begin{aligned}
CV(h_1,h_2) =& \frac{1}{n^2h_1h_2}\sum_{i=1}^{n}\sum_{j=1}^{n}K*K\left(\frac{x_j-x_i}{h_1},\frac{y_j-y_i}{h_2}\right) \\
&- \frac{2}{n(n-1)h_1h_2}\sum_{i=1}^{n}\sum_{j\neq i}K\left(\frac{x_j-x_i}{h_1},\frac{y_j-y_i}{h_2}\right)
\end{aligned} \tag{C.50}$$

We choose $h_1$ and $h_2$ which minimizes $CV(h_1, h_2)$. If $K(x, y)$ is the probability density function of the product Gaussian Kernel, which is $N(\mathbf{0}, \mathbf{I}_2)$, where $\mathbf{I}_2$ denotes the $2 \times 2$ identity matrix, then $K * K$ is the probability density function of a $N(\mathbf{0}, 2\mathbf{I}_2)$ random variable.

## C.12 Unbiased Cross-Validation for Gaussian Kernels

In the general case

$$\begin{aligned} UCV(\mathbf{H}) =& \frac{1}{n^2 \det(\mathbf{H})} \sum_{i=1}^{n} \sum_{j=1}^{n} K * K(\mathbf{H}^{-1}(\mathbf{x}_j - \mathbf{x}_i)) \\ & - \frac{2}{n(n-1) \det(\mathbf{H})} \sum_{i=1}^{n} \sum_{j \neq i} K(\mathbf{H}^{-1}(\mathbf{x}_j - \mathbf{x}_i)) \end{aligned} \tag{C.51}$$

If $\mathbf{H}$ is a diagonal $2 \times 2$ matrix $\left[\begin{smallmatrix} h_1 & 0 \\ 0 & h_2 \end{smallmatrix}\right]$, then:

$$\begin{aligned} UCV(h_1, h_2) =& \frac{1}{n^2 h_1 h_2} \sum_{i=1}^{n} \sum_{j=1}^{n} K * K \left( \frac{x_j - x_i}{h_1}, \frac{y_j - y_i}{h_2} \right) \\ & - \frac{2}{n(n-1) h_1 h_2} \sum_{i=1}^{n} \sum_{j \neq i} K \left( \frac{x_j - x_i}{h_1}, \frac{y_j - y_i}{h_2} \right) \end{aligned} \tag{C.52}$$

Assuming $K$ is the pdf of a $N(\mathbf{0}, \mathbf{I}_2)$ random variable, i.e.:

$$K(x, y) = K(x)\, K(y) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{x^2}{2} \right\} \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{y^2}{2} \right\} = \frac{1}{2\pi} \exp\left\{ -\frac{x^2 + y^2}{2} \right\} \tag{C.53}$$

the convolution $K * K$ is the pdf of a $N(0,\, 2I_2)$ random variable, i.e.:

$$K * K(x, y) = \frac{1}{\sqrt{2\pi}\sqrt{2}} \exp\left\{ -\frac{x^2}{2(\sqrt{2})^2} \right\} \frac{1}{\sqrt{2\pi}\sqrt{2}} \exp\left\{ -\frac{y^2}{2(\sqrt{2})^2} \right\} = \frac{1}{4\pi} \exp\left\{ -\frac{x^2 + y^2}{4} \right\} \tag{C.54}$$

Thus, the formula for the unbiased cross validation objective function becomes:

$$\begin{aligned} UCV(h_1,h_2) =& \frac{1}{n^2 h_1 h_2} \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{4\pi} \exp\left\{-\frac{1}{4}\left[\left(\frac{x_j - x_i}{h_1}\right)^2 + \left(\frac{y_j - y_i}{h_2}\right)^2\right]\right\} \\ & - \frac{2}{n^2 h_1 h_2} \sum_{i=1}^{n}\sum_{j\neq i} \frac{1}{2\pi} \exp\left\{-\frac{1}{2}\left[\left(\frac{x_j - x_i}{h_1}\right)^2 + \left(\frac{y_j - y_i}{h_2}\right)^2\right]\right\} \end{aligned} \tag{C.55}$$

If we denote by

$$T_{ij} = \exp\left\{-\frac{1}{4}\left[\left(\frac{x_j - x_i}{h_1}\right)^2 + \left(\frac{y_j - y_i}{h_2}\right)^2\right]\right\} \tag{C.56}$$

then we have:

$$\exp\left\{-\frac{1}{2}\left[\left(\frac{x_j - x_i}{h_1}\right)^2 + \left(\frac{y_j - y_i}{h_2}\right)^2\right]\right\} = T_{ij}^2 \tag{C.57}$$

If $i = j$ then:

$$T_{ij} = \exp\left\{-\frac{1}{4}\left[\left(\frac{x_j - x_i}{h_1}\right)^2 + \left(\frac{y_j - y_i}{h_2}\right)^2\right]\right\} = e^0 = 1 \tag{C.58}$$

and we also have $T_{ij} = T_{ji}$, therefore it is sufficient to calculate $T_{ij}$ only for $i < j$. Thus, the formula for unbiased cross-validation becomes:

$$
\begin{aligned}
UCV(h_1, h_2) &= \frac{1}{4\pi n^2 h_1 h_2} \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} - \frac{1}{\pi n^2 h_1 h_2} \sum_{i=1}^{n} \sum_{j \neq i} T_{ij}^2 \\
&= \frac{1}{4\pi n^2 h_1 h_2} \sum_{i=1}^{n} \left( 1 + 2 \sum_{j>i} T_{ij} \right) - \frac{1}{\pi n^2 h_1 h_2} \sum_{i=1}^{n} \left( 2 \sum_{j>i} T_{ij}^2 \right) \\
&= \frac{1}{4\pi n^2 h_1 h_2} \left( n + 2 \sum_{i=1} \sum_{j>i} T_{ij}^2 \right) \qquad \text{(C.59)} \\
&= \frac{1}{4\pi n h_1 h_2} + \frac{1}{2\pi n^2 h_1 h_2} \left( \sum_{i=1} \sum_{j>i} T_{ij} - 4 \sum_{i=1} \sum_{j>i} T_{ij}^2 \right) \\
&= \frac{1}{4\pi n h_1 h_2} + \frac{1}{2\pi n^2 h_1 h_2} \sum_{i=1} \sum_{j>i} \left[ T_{ij} \left( 1 - 4 T_{ij} \right) \right]
\end{aligned}
$$

# Bibliography

# Bibliography


[1] I. Poese, S. Uhlig, M. Kaafar, B. Donnet, and B. Gueye, "IP geolocation databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53–56, 2011.

[2] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for Internet hosts," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 173–185, Oct. 2001.

[3] A. Ziviani, S. Fdida, J. de Rezende, and O. Duarte, "Toward a measurement-based geographic location service," in *Proc. of Passive and Active Network Measurement (PAM)*, Antibes Juan-les-Pins, 2004, pp. 43–52.

[4] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," in *Proc. 4th ACM SIGCOMM Conf. on Internet measurement*, 2004, pp. 288–293.

[5] ——, "Constraint-based geolocation of Internet hosts," *IEEE/ACM Trans. on Networking*, vol. 14, no. 6, pp. 1219–1232, 2006.

[6] E. Katz-Bassett, J. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP geolocation using delay and topology measurements," in *Proc. 6th ACM SIGCOMM Conf. on Internet measurement*, 2006, pp. 71–84.

[7] I. Youn, B. L. Mark, and D. Richards, "Statistical Geolocation of Internet Hosts," *Proc. 18th Int. Conf. on Computer Communications and Networks (ICCCN)*, pp. 1–6, Aug. 2009.

[8] M. Arif, S. Karunasekera, and S. Kulkarni, "GeoWeight: internet host geolocation based on a probability model for latency measurements," in *Proc. 33rd Australasian Conf. on Computer Science*, vol. 102, 2010, pp. 89–98.

[9] Creativerge, "Multilateration and ADS-B. Executive Reference guide," http://www.multilateration.com.

[10] R. Bucher and D. Misra, "A synthesizable VHDL model of the exact solution for three-dimensional hyperbolic positioning system," *Vlsi Design*, vol. 15, no. 2, pp. 507–520, 2002.

[11] M. J. Arif, S. Karunasekera, S. Kulkarni, A. Gunatilaka, and B. Ristic, "Internet Host Geolocation Using Maximum Likelihood Estimation Technique," *24th IEEE Int. Conf. on Advanced Information Networking and Applications*, pp. 422–429, 2010.

[12] B. Wong, I. Stoyanov, and E. Sirer, "Octant: A comprehensive framework for the geolocalization of Internet hosts," in *Proc. NSDI*, 2007.

[13] F. Pérez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions," in *IEEE Int. Symp. Information Theory (ISIT)*, 2008, pp. 1666–1670.

[14] L. Orlóci, "An agglomerative method for classification of plant communities," *J. Ecology*, pp. 193–205, 1967.

[15] L. L. Cavalli-Sforza and A. W. F. Edwards, "Phylogenetic analysis: models and estimation procedures," *Evolution*, vol. 21, pp. 550–570, 1967.

[16] P. Legendre and E. Gallagher, "Ecologically meaningful transformations for ordination of species data," *Oecologia*, vol. 129, no. 2, pp. 271–280, 2001.

[17] C. R. Rao, "A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance," *Qüestiió*, vol. 19, pp. 23–63, 1995.

[18] P. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of the National Institute of Sciences of India*, vol. 2, no. 1. New Delhi, 1936, pp. 49–55.

[19] G. Lance and W. Williams, "Computer programs for hierarchical polythetic classification (similarity analyses)," *The Computer Journal*, vol. 9, no. 1, pp. 60–64, 1966.

[20] A. Gordon, *Classification: Methods for the Exploratory Analysis of Multivariate Data*, ser. Monographs on Statistics and Applied Probability, 82. Chapman & Hall, 1999.

[21] P. Clark, "An extension of the coefficient of divergence for use with multiple characters," *Copeia*, pp. 61–64, 1952.

[22] PlanetLab, http://www.planet-lab.org.

[23] B. Silverman, *Density Estimation*, ser. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1986.

[24] D. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, ser. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1992.

[25] W. Härdle, *Nonparametric and Semiparametric Models*, ser. Springer Series in Statistics Series. Springer-Verlag GmbH, 2004.

[26] Geospatial Science Division, "World Geodetic System 1984," U.S. Dept. of Defense, Tech. Rep. TR8350.2, July 1997, 3rd Ed.

[27] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey review*, vol. 23, no. 176, pp. 88–93, 1975.

[28] C. Veness, "Vincenty Direct and Inverse Solutions of Geodesics on the Elipsoid," http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html.

[29] S. Sain, K. Baggerly, and D. Scott, "Cross-validation of multivariate densities," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 807–817, 1994.

[30] D. Scott and G. Terrell, "Biased and unbiased cross-validation in density estimation," *Journal of the American Statistical Association*, pp. 1131–1146, 1987.

[31] M. Wand and C. Jones, *Kernel Smoothing*, ser. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1995.

# Curriculum Vitae

Inja Youn finished her Master's degree in Computer Science at George Mason University in 2004. Her research areas of interest include Internet measurements, cryptography, network security, high-performance computing and scientific programming.