CHALMERS

A Survey of Information Geometry in Iterative Receiver Design, and Its Application to Cooperative Positioning

Master of Science Thesis in Communication Engineering

DAPENG LIU

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2012
Report No. xxxx

REPORT NO. XXX/XXXX

A Survey of Information Geometry in Iterative Receiver Design, and Its Application to Cooperative Positioning

Thesis for the degree of Master of Science

DAPENG LIU

CHALMERS

Department of Signals and Systems

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2012

A Survey of Information Geometry in Iterative Receiver Design, and Its Application to Cooperative Positioning

DAPENG LIU

Technical report no. xxxxx/xxxx

Department of Signals and Systems

Chalmers University of Technology

SE-412 96 Gothenburg

Sweden

Abstract

Belief propagation is a powerful procedure to perform inference, and exhibits near-optimal performance in iterative decoding/demapping. However, a complete analysis of the iterative process is still lacking, and convergence of the procedure has not proved in general cases. Information geometry offers a new view from differential geometry to understand this problem. In this thesis, we try to understand belief propagation as an iterative projection procedure by studying various information-geometric interpretations from the literature. We extend these insights to distributed inference, in particular to cooperative positioning. Cooperative Bayesian algorithms have outstanding performance in many network scenarios. However, they suffers from a large computational complexity when messages are represented by a grid. We propose three techniques to reduce the complexity and improve the accuracy of cooperative Bayesian positioning: geometrical pre-location, dynamic grid, and clipping. The performance of the new algorithm compares favorably with the original algorithm, with considerably complexity and network traffic reduction.

Keywords: Belief propagation, Information geometry, Iterative receiver, Cooperative positioning, Complexity and network traffic reduction.

Contents

List of Algorithms

Overall Introduction

This is a two-part thesis. The first part is a literature overview on the interpretation of belief propagation as iterative projection procedure in information geometry. The second part uses the knowledge learned from the first part and applies it to the problem of cooperative localization.

In Part I, we first introduce some basic concepts and knowledge about information geometry regarding the requirements for the further discussion. Later, we try to offer a general picture of the current interpretations and algorithms by the analysis and comparison of representative former works. At last we draw the conclusions and summary.

In Part II, a new cooperative positioning algorithm is proposed based on the cooperative Bayesian algorithms. Since the former algorithm is heavily depending on belief propagation, we can easily apply the understanding got from the first part to solve the problem here. Combining a few novel ideas, the new algorithm largely reduces the complexity and also gains many other advantages.

Though the fields of information geometry and cooperative localization at the first glance seem not so related, it is the study of belief propagation and its application that unites the two parts as a complete thesis.

Part I

Information Geometric Interpretation of Belief Propagation

Chapter 1

Introduction

The belief propagation algorithm has been invented over twenty years as a powerful estimation procedure, which performs inference by message passing on graphical models [1]. The algorithm might be initialized as an efficient method to calculate marginal probabilities when the structure of factor graphs is forest, yet its amazing performance lies in the successful application in factor graphs with loops [2]. When applied to the iterative decoding/demapping process of Turbo codes, LDPC codes, and BICM, people realize its importance as the practical validation occurs [3]. However, a complete analysis of the iterative process is still lacking, and convergence of belief propagation procedure has not been proved in general cases.

Information geometry, on another hand, as a rising star offers a brand new view of belief propagation and inference theory from differential geometry [4]. With the foundation stone, the work of Shun'ichi Amri, information geometry is becoming a systematized and matured subject. The main principle of this relatively young theory is to study information theory, probability theory, and statistics by rephrasing many basic concepts as the structures in differential geometry.

Recently, a growing attention has risen on applying information geometry to interpret the iterative process of belief propagation by iterative information projections [5]. Although aim-

ing for the same purpose, many work have taken very different paths which reveal different aspects.

In this thesis, by comparing and combining various information-geometric interpretations from former works, the intent is to rephrase belief propagation as iterative projection procedure, present the analysis and fully comparison of each important information projection algorithms. Meanwhile, by comparing the process of information projection algorithms and belief propagation on factor graphs, we hope to gain fresh understanding on iterative receivers and extend these insights to distributed inference, in particular to cooperative positioning.

The rest part of the thesis is organized as follow,

- Chapter 2 Basic knowledge of information geometry

  In this chapter, basic concepts of information geometry is introduced, including families of distributions, Kullback-Leibler divergence and more generally, Bregman divergence, at last information projections both left and right projections.

- Chapter 3 Analysis and comparison of important former work.

  Four representative works are chosen to demonstrate from different aspects and levels of how iterative decoding/demapping and belief propagation are rephrased as iterative information projections.

- Chapter 4 Conclusion and future work

  We discuss and summarize the contributions and shortcomings of each work.

Chapter 2

Basic Knowledge of Information Geometry

In this chapter, we present the several fundamental concepts in information geometry, and convex projections, mainly based on the book, methods of information geometry, by S. Amari [4], the master thesis by Stefan Schwandter [6], and the paper by Walsh and Regalia [7].

Information geometry is a relatively new method which emerged from the investigation of the geometrical structures of manifolds formed by probability distributions. By borrowing the concepts from differential geometry, we can explore the geometric properties of probability and information which offers many interesting interpretations of the original knowledge in geometrical way.

Although information geometry has attractive utility in plenty of fields, it is not easy to master. However, in this thesis, it only requires some essential knowledge

2.1 Manifolds and Important Families of Distributions

Manifolds, as families of probability distributions, are the fundamental concepts in information geometry. A family of probability distribution is a set of distributions with the same structure characterized by a group of parameters. Combinations of parameters with arbitrary values form various distributions, yet all the distributions follow a particular structure. Each distribution in a family can be visualized as a point in a manifold. Because it is one-to-one mapping between a combination of parameters and a distribution, the parameters can serve as the coordinates of the family or the manifold.

Exponential Families

Exponential families have two alternative definitions. The common definition of an exponential family:

Definition 2.1.1 the set $\mathscr{E}$ for all discrete probability distribution $p(x), x \in \mathscr{X}$ that satisfy:

$$\mathscr{E} = \left\{ p \mid p(x) = cq(x)exp(\sum_{i=1}^{n} \theta_i f_i(x)) \right\} \tag{2.1}$$

where c is a normalization factor $c = (\sum q(x)exp(\sum_{i=1}^{n} \theta_i f_i(x)))^{-1}$, $q(x)$ is a given distribution, $\theta_1, \theta_2 ... \theta_n$ are scalars and $f_1, f_2 ... f_n$ are given functions on $\mathscr{X}$. Specially, all these f_i, $i = 1, 2, \cdots n$ functions together called "sufficient statistics". As can be seen from the formula, the whole family is built around the given distribution $q(x)$, and θ_i, $i = 1, 2, \cdots n$ in vector θserves as the coordinate system of the family. Though one can choose other alternate coordinate systems, the vector $\theta$is named as "natural coordinate".

Second definition of an exponential family is more focusing on one of its important geometric property:

An exponential family is a set of discrete probability distributions with e-geodesic (expo-

nential geodesic) between any two of its elements.

Assume two distributions $p(x)$ and $q(x)$ belong to a submanifold M in a exponential family $\mathscr{E}$, $\exists r(x;t) \in M$ for all $t \in [0, 1]$,

$$\ln r(x;t) = t \ln p(x) + (1-t) \ln q(x) + c(t) \tag{2.2}$$

In exponential domain, M is a one-dimensional curve. If with natural coordinates θ, it then appears as a "straight line", $\theta_r = t\,\theta_p + (1-t)\,\theta_q$ thus it is an e-geodesic between pointθ_p and θ_q. In some works, it is mentioned as e-flat submanifold.

Linear Families

For any given function $f_1, f_2 \ldots f_n$ on $\mathscr{H}$ and numbers $\alpha_1, \alpha_2 \ldots \alpha_n$, a linear family of probability distributions is the set

$$\mathscr{L} = \{p : \sum_x f_i(x)\, p(x) = \alpha_i,\ 1 \le i \le n\}. \tag{2.3}$$

Namely, the number α_i can be considered as the expectation value of $f_i(x)$ regarding the distribution $p(x)$ with the random variable x. Assume vector $\alpha$and $f(x)$ collects all α_i and $f_i(x)$, $i \in [1, n]$, then the definition can be reformed as

$$\mathscr{L} = \{p : \mathbb{E}_p\{f(x)\} = \alpha \tag{2.4}$$

Vector α is also a coordinate system, called "mixture coordinates" for the linear family. Compare to exponential family, linear families has another definition which reveals its property, too.

Similarly, submanifold $M \subset \mathscr{L}$, for all $t \in [0, 1]$, $p(x)$, $q(x) \in M$, the following mixture

$r(x;t)$ belongs to M,

$$r(x;t) = t\,p(x) + (1-t)\,q(x). \tag{2.5}$$

Submanifold M is also a one-dimensional curve appearing to be straight line in mixture coordinates, thus it is called "m-geodesic". In some papers, it is also mentioned as m-flat submanifold.

2.2 Kullback-Leibler Divergence, an Instance of Bregman Divergence

In any kind of geometry, the distance measurement can be seen as one of the most important foundation stone. Considering information geometry, Kullback-Leibler divergence serves this role as the basic metric, the "distance" between any two points formed by probability distributions.

Definition 2.2.1 For two probability distributions p and q of a discrete random variable x on $\mathscr{X}$, Kullback-Leibler (KL) divergence is defined as

$$D(p \parallel q) = \sum_{\mathscr{X}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})}. \tag{2.6}$$

Namely, it shows the expectation of function $f(x) = \ln \frac{p(x)}{q(x)}$, the logarithmic difference between the two distribution p and q. Note that the expectation is with probability distribution $p(x)$ and if $0 \ln 0$ appears, we interpreted it as 0.

As other normal distance metric, KL divergence is always non-negative:

$$D(p \parallel q) \geq 0.$$

And $D(p \parallel q) = 0$, if and only if $p = q$. It is also additive for independent distributions. For

instance, $p = p_1(x)p_2(y)$ and $q = q_1(x)q_2(y)$, then

$$D(p \parallel q) = .D(p_1 \parallel q_1) + D(p_2 \parallel q_2)$$

However, there are a few notable properties that we should pay attention. First, it is not symmetric, namely, $D(p \parallel q) \neq D(q \parallel p)$. Secondly, it does not satisfy the triangle inequality. These properties make KL divergence not a metric intuitively.

Generally, KL divergence can been seen as a special instance of Bregman Divergence. Assume $f(\mathbf{x})$ is a convex function defined over a convex domain $\mathscr{D} \subseteq \mathbb{R}^N$. As $f(\cdot)$ is convex, assume it is differentiable on domain $\mathscr{D}$, so the tangent hyperplane of function $f(\mathbf{x})$ at any point$\mathbf{x}$ reveals its lower bound. This lead us to the gradient inequality

$$f(\mathbf{y}) \geq f(\mathbf{x}) + < \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} >, \tag{2.7}$$

where $\nabla f(\mathbf{x}) = df(\mathbf{x})/d\mathbf{x}$ is the gradient vector, and $< \cdot, \cdot >$stands for the standard inner product. When $f(\mathbf{x})$ is strictly convex and differentiable, the Bregman divergence is defined as

$$D_f(\mathbf{y}, \mathbf{x}) = f(\mathbf{y}) - f(\mathbf{x}) - < \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} > . \tag{2.8}$$

Since it is strictly convex, the Bregman divergence $D_f(\mathbf{y}, \mathbf{x}) = 0$, if and only if $\mathbf{y} = \mathbf{x}$.

For instance, Euclidean Distance Squared can be seen as an example of Bregman divergence. Suppose function

$$f(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{N} x_i^2$$

on domain$\mathscr{D} \subseteq \mathbb{R}^N$, then the Bregman divergence is

$$D_f(\mathbf{y}, \mathbf{x}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - x_i)^2.$$

So when $\mathbf{x} = [x_1, \ldots x_{N-1}]$ and $x_N = 1 - \sum_{i=1}^{N-1} x_i$, the function now is negative Shannon entropy

$$f(\mathbf{x}) = \sum_{i=1}^{N} x_i \log x_i,$$

and it is convex within domain $\mathscr{D} = \{\mathbf{x} | x_i \geq 0, \forall i \in \{1, \ldots N-1\}, \sum_{i=1}^{N-1} x_i \leq 1\}$. This function induced the Bregman divergence is

$$D_f(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{N} y_i \log \frac{y_i}{x_i}.$$

For every convex function $f(\mathbf{x})$, there is a conjugate function $f^*(\theta)$, respectively, which is also convex . $f^*(\theta)$ is defined as

$$f^*(\theta) = \sup_{\mathbf{x}}(< \mathbf{x}, \theta > -f(\mathbf{x})) \tag{2.9}$$

If the function f is strictly convex and differentiable everywhere inside the domain $\mathscr{D}$, where $\mathscr{D}$ is convex and open with $\nabla f(\mathbf{x}^i) \to \infty$ for any sequence $\mathbf{x}^i$ approaching a boundary [8], then this function is called Legendre type. Under this conditions, function f and f^* is called a Legendre transform pair [8] and the gradients $\nabla f(\mathbf{x})$ and $\nabla f^*(\theta)$ are inverse maps to each other.

The conjugate function to the negative Shannon entropy is the log partition function, i.e., with $f(\mathbf{x}) = \sum_{i=1}^{N} x_i \log x_i$,

$$f^*(\theta) = \sup_{\mathbf{x}}(< \mathbf{x}, \theta > -f(\mathbf{x})) = \log(1 + \sum_{i=1}^{N-1} \exp(\theta_i)), \tag{2.10}$$

where $\theta = [\theta_1, \ldots, \theta_{N-1}]^T$, $\theta_i = \log \frac{x_i}{x_N}$.Now the gradients ∇f and $\nabla f^*$are inverses to each other, as they are a Legendre transform pair.

If function f is Legendre type, the conjugate function f^* can be reformed as

$$\begin{aligned} f^*(\theta) &= \sup(<\mathbf{x}, \theta> - f(\mathbf{x})) \\ &= <(\nabla f)^{-1}(\theta), \theta> - f((\nabla f)^{-1}(\theta)) \\ &= <\nabla f^*(\theta), \theta> - f(f^*(\theta)). \end{aligned}$$

With this deduction, Bregman divergence can also be written as

$$\begin{aligned} D_{f*}(\rho,\theta) &= f^*(\rho) - f^*(\theta) - <\nabla f^*(\theta), \rho - \theta> \\ &= <\nabla f^*(\rho), \rho> - f(f^*(\rho)) - <\nabla f^*(\theta), \theta> + f(f^*(\theta)) \\ &\quad - <\nabla f^*(\theta), \rho - \theta> \\ &= f(\nabla f^*(\theta)) - f(\nabla f^*(\rho)) - <\rho, \nabla f^*(\theta) - \nabla f^*(\rho)> \\ &= D_f(\nabla f^*(\theta), \nabla f^*(\rho)). \end{aligned}$$

So as can be seen above, if we assume $\rho = \nabla f(\mathbf{y})$ and $\theta = \nabla f(\mathbf{x})$, then we have Bregman divergence with switch arguments between D_f and D_{f^*},

$$D_f(\mathbf{y}, \mathbf{x}) = D_{f^*}(\theta, \rho) \tag{2.11}$$

2.3 Information Projections

With the distance metric, the Kullback-Leibler divergence as an instance of Bregman divergence, now it is able to define a basic behavior in geometry, the projections. As mentioned above, since KL divergence is not symmetric, i.e., $D(p \parallel q) \neq D(q \parallel p)$, we can define the two different projections represent distance measured with each way. In general, the projections reveal the closest points inside a manifold to a given point outside.

I-projection

Definition 2.3.1 Given a distribution q and a closed convex subset $\mathscr{D}$ of distribution $\mathscr{A}$, the I-projection of q onto $\mathscr{D}$ is a distribution $p' \in \mathscr{D}$ that

$$D(p'||q) = \min_{p \in \mathscr{D}} D(p||q). \tag{2.12}$$

In many works [9], the I-projection is also mentioned as e-projection, as the curve linking p' and q is an e-geodesic. Analogous to Pythagorean theorem in Euclidean geometry, a similar theorem is also valid in information geometry,

$$D(p||q) \geq D(p||p') + D(p'||q), \forall p \in \mathscr{D}$$

The equality is achieved when subset $\mathscr{D}$ is a linear family.

Theorem 2.3.1 The I-projection of q onto a linear family $\mathscr{L}$ is unique and satisfies the Pythagorean identity

$$D(p||q) = D(p||p') + D(p'||q), \forall p \in \mathscr{L} \tag{2.13}$$

Since it is a one-dimensional exponential family that connects p' and q [9], the distribution p' can be written in two ways, both in exponential family $\mathscr{E}_q$ and in linear family $\mathscr{L}$.

As I-projection is from q, the right hand side in $D(p||q)$, onto p which is on the left, so in some cases we use this symbol $\overset{\mathscr{S}\leftarrow}{\pi}_f(q)$ to denotes the I-projection from q to the manifold $\mathscr{S}$ with the function f used in Bregman divergence, namely,

$$p' = \overset{\mathscr{S}\leftarrow}{\pi}_f(q) = \arg\min_{p \in \mathscr{S}} D_f(p||q). \tag{2.14}$$

Reverse I-projection

As I-projection takes p in submanifold as a variable in $D(p||q)$, the reverse I-projection, on the other hand, keeps p fixed and varying q to optimized the solution.

Definition 2.3.2 Given a distribution p and a closed convex subset $\mathscr{D}$ of distributions on $\mathscr{A}$, the reverse I-projection (rI-projection) of p onto $\mathscr{D}$ is a distribution $q' \in \mathscr{D}$ that

$$D(p||q') = \min_{q \in \mathscr{D}} D(p||q). \tag{2.15}$$

The reverse I-projection is also named m-projection, for the curve linking p and q' appeals a straight line as m-geodesic.

Similarly, there exists a Pythagorean identity of reverse I-projections, when subset $\mathscr{D}$ is en exponential family.

Theorem 2.3.2 The reverse I-projection of p onto an exponential family $\mathscr{E}$ is unique and satisfies the Pythagorean identity

$$D(p||q) = D(p||q') + D(q'||q), \forall q \in \mathscr{E}. \tag{2.16}$$

Compared to formula 2.14, here we use this symbol $\overrightarrow{\pi}_f^{\mathscr{S}}(p)$ to denotes the rI-projection from p to the manifold $\mathscr{S}$ with the function f used in Bregman divergence, namely,

$$q' = \overrightarrow{\pi}_f^{\mathscr{S}}(p) = \arg\min_{q \in \mathscr{S}} D_f(p||q). \tag{2.17}$$

Projections on Factorisable Distributions

As factorisable distributions are very important for decoding process, so here we discuss some properties when projecting a given distribution on to factorisable submanifold by either I-projection or rI-projection.

For I-projections, when the subset which projected on is a linear family $\mathscr{L}_{\mathscr{C}}$, using notation defined in definition 2.3.1, the p' of the I-projection of q can be written in such form,

$$p'(\mathbf{x}) = \arg\min_{p \in \mathscr{L}_{\mathscr{C}}} D(p||q) = q(\mathbf{x}) \exp(-\sum_{1}^{r} \mu_i (f_i(\mathbf{x}) - \alpha_i)) \tag{2.18}$$

where the set$\{\mu_i\}$ are Lagrange multipliers determined from the constraints [10]. When the functions set $\{f_i\}$ of the linear family are specially designed as the parity-check equations of a code and in some particular case where $\{\alpha_i\} = 0$, then the equation 2.18 can be reformed as

$$p'(\mathbf{x}) = q(\mathbf{x}) \exp(-\mu_0) \prod_{1}^{r} I_i(\mathbf{x}) \tag{2.19}$$

where $\exp(-\mu_0)$ is for normalization and $I_i(\mathbf{x})$ is the indicator function for all vectors $\mathbf{x}$ which satisfy constraint f_i. Because a codeword has to follow all the parity-check constraints at the same time, we can indicate a codeword with $I_{\mathscr{C}} = \prod_{1}^{r} I_i(\mathbf{x})$. Therefore, the I-projection of q onto the linear family $\mathscr{L}_{\mathscr{C}}$ with the parity-check equations of code $\mathscr{C}$ can be written below:

$$p'(\mathbf{x}) = \frac{q(\mathbf{x}) I_{\mathscr{C}}(\mathbf{x})}{\sum_{\mathbf{x} \in \mathscr{X}} q(\mathbf{x}) I_{\mathscr{C}}(\mathbf{x})} \tag{2.20}$$

For rI-projection, when project p onto a factorisable exponential family $\mathscr{E}_f = \{p(\mathbf{x}) | \, p(\mathbf{x}) = \prod_{i}^{r} p_i(x_i)\}$, $q' \in \mathscr{E}_f$ is the product of the marginal probabilities of $\mathbf{x}$, which is

$$q'(\mathbf{x}) = \arg\min_{q \in \mathscr{E}_f} D(p||q) = \prod_{i}^{r} p_i(x_i). \tag{2.21}$$

Here $p_i(x_i) = \sum_{\mathbf{x} \sim x_i} p(\mathbf{x})$, it is the marginal distribution on x_i. We can simply demo the proof

below:

$$\begin{aligned} D(p||q) &= \sum_{\mathscr{X}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})}. \\ &= \sum_{\mathscr{X}} p(\mathbf{x}) (\log \frac{p(x)}{\prod_i p_i(x_i)} + \log \frac{\prod_i p_i(x_i)}{\prod_i q_i(x_i)}) \\ &= D(p||p') + \sum_{\mathscr{X}} p(\mathbf{x}) \sum_i \log \frac{p_i(x_i)}{q_i(x_i)}) \\ &= D(p||p') + \sum_i \sum_{x_i} \sum_{\mathbf{x} \sim x_i} p(\mathbf{x}) \log \frac{p_i(x_i)}{q_i(x_i)} \\ &= D(p||p') + \sum_i D(p_i||q_i). \end{aligned}$$

As we know, the KL divergence is never negative, and $D(p_i||q_i) = 0$ only when $p_i = q_i$, so follow the definition, the rI-projection of p onto $\mathscr{E}_f$ is p' with the marginal distribution on x_i.

Chapter 3

Analysis on Representative Papers

In this chapter, we introduce four representative papers in information geometrical interpretation of iterative decoding. All the four papers are based on the common fundamental concepts and theorems discussed in Chapter 2, however, each paper has its own emphasis and different aspect of view. Here we may only focus on those parts of the papers which interest us most and abandon many contents in the original papers. It is worthy to mention that the four papers contain many different notations and symbols, while in this thesis, we try to keep them consistent, which means, we may use different symbols to present the same thing in the original papers. The paper we choose are listed below:

- Paper A [6]: Stefan Schwandter, "Information Geometric Analysis of Iterative Receiver Structures", *Master thesis, Vienna University of Technology*, 2008

- Paper B [11]: Florence Alberge, "Iterative decoding as Dykstra's algorithm with alternate I-projection and reverse I-projection", *presented at the 16th European Signal Processing Conf. (EUSIPCO)*, 2008

- Paper C [7]: John Walsh, "Belief Propagation, Dykstra Algorithm, and Iterated Information Projections", *IEEE, Trans. Inf. Theory*, vol. 56, no. 8, pp. 4114–4128,

2010

- Paper D [12]: S. Ikeda, T. Tanaka, and S. Amari, "Stochastic Reasoning, Free Energy and Information Geometry", *Neural Computation*, vol. 16, no. 9, pp. 1779–1810, 2004

The first paper, information geometric analysis of iterative receiver structures, is a master thesis written by Stefan Schwandter from Vienna University of Technology, 2008. This paper contains detailed examples to help understanding, very useful for those who are new to this field. The second paper is iterative decoding as Dykstra's algorithm with alternate I-projection and reverse I-projection by Florence Alberge [11]. In the paper, the author rephrases the belief propagation as an iterative projections process onto variable manifolds. The third paper is belief propagation, dykstra algorithm, and iterated information projections, proposed by John Walsh [7]. Here the author offers a new way of projections to get rid of the changeing manifolds in the second paper. The last paper we choose is from birth place of information geometry by S. Ikeda, T. Tanaka, and S. Amari, stochastic reasoning, free energy and information geometry [12]. This paper can help the reader visualize the process of belief propagation in information geometry and it focuses on the convergence condition of belief propagation.

3.1 Paper A: Information Geometric Analysis of Iterative Receiver Structures

To continue with what we presented in Chapter 2, this paper (Paper A) introduces the algorithms of iterative projections and later to interpret the general soft sequence decoding process with the iterative projections.

3.1.1 Iterative Projections Algorithms

It is a natural interests that how to find the minimum KL divergence between any given two convex sets of distributions. Let $\mathscr{P}$ and $\mathscr{Q}$ be the two sets of discrete probability distributions, both convex on a finite alphabet $\mathscr{X}$, the KL divergence $D(p||q)$ should be minimized with two distributions p_{final} and q_{final} which gives,

$$\begin{aligned} D_{\min} &= \min_{p\in\mathscr{P}, q\in\mathscr{Q}} D(p||q) \\ &= D(p_{final}||q_{final}). \end{aligned}$$

Suppose the iterative projections starts from a certain distribution $q_0 \in \mathscr{Q}$, the next distribution $p_1 \in \mathscr{P}$ is achieved by I-projection of q_0 onto $\mathscr{P}$. In other the way, we can simply induct as

$$p_n = \overleftarrow{\pi}_f^{\mathscr{P}}(q_{n-1}),$$

and $q_n \in \mathscr{Q}$ is the rI-projection of p_n onto $\mathscr{Q}$,

$$q_n = \overrightarrow{\pi}_f^{\mathscr{Q}}(p_n).$$

With a general theorem proved in [13], the iterative projections converge when n goes infinite,

$$D(p_n||q_n) \to D_{\min}, n \to \infty.$$

If the set $\mathscr{P}$ is compact, p_n converges to an p_∞, such that

$$D(p_\infty||\overrightarrow{\pi}_f^{\mathscr{Q}}(p_\infty)) = D_{\min}.$$

With the same condition, the iterations can be stopped on the inequality

$$D(p_{n+1}||q_n) - D_{\min} \leq \max_{\mathbf{x}\in\mathscr{X}} \log \frac{p_{n+1}}{p_n}.$$

3.1.2 EM Algorithm

After introduced the algorithm of iterative projections, the author demonstrates a useful instance in statistics domain to estimate a probability distribution, which is ought to be in a set of distributions, when only incomplete sample data available.

Assume $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$ is the original sample, and $\mathbf{y} = (y_1, y_2, \ldots, y_N) \in \mathscr{Y}^N$ is the observed incomplete sample, with the mapping $T : \mathscr{X} \mapsto \mathscr{Y}$, $y_i = Tx_i$. Usually, the mapping T is a many-to-one mapping, in another word, there are information lost during the mapping. One example can be a histogram that convert many distinguish value into a single one.

The process of the EM algorithm is by updating the distribution $q_n \in \mathscr{Q}$ to improve the estimations of the unknown distribution. It begins with a arbitrary distribution q_0 and repeats two steps consecutively, E-step and M-step.

E-step

E-step is to calculate the conditional expectations of an empirical distribution$\hat{p_N}$ [14], from the original sample, given the observed distributions $\mathbf{y}$,

$$p_n = \mathbb{E}_{q_{n-1}}\{\hat{p_N}|\mathbf{y}\},$$

where $\hat{p_N} = \frac{number\,of\,occurrences\,of\,a\,in\,sequence\,\mathbf{x}}{N}$. As the expectation is carried out with the current estimation of the true distribution q_{n-1}, the result of the E-step, distribution p_n, fuses with the information from current estimation and the observed samples.

M-step

After E-step, M-step is to calculated the maximum-likelihood estimation from the empirical distribution p_n to a distribution$q_n$which belong to the set $\mathscr{Q}$where the true distribution lies in. This ML estimate is achieved through minimizing the KL divergence [13], namely

$$q_n = \arg\min_{q \in \mathscr{Q}} D(p_n||q) = \overrightarrow{\pi}_f^{\mathscr{Q}}(p_n).$$

It is easily to see that the M-step is a rI-projection of p_n onto set $\mathscr{Q}$, however, it can also be proved that E-step is I-projection of q_{n-1}onto a set $\mathscr{P}$ where $\mathscr{P} = \{p : p^T = \hat{p_N}\}$.

3.1.3 Geometrical Interpretation of Soft Sequence Decoding

Generally, the optimum decoder (in terms of sequence error probability) of BICM, LDPC code or turbo code, would be the maximum a posterior (MAP) decoder, because it chooses the sequence with the highest probability through all possible combinations. However, the computational complexity of this decoder goes exponentially with the code length. Some proposed algorithms can reduce the complexity, yet demanding of a memoryless channel, which cannot be fulfilled due to the bit interleavers.

Therefore, it is worthy to analysis soft decoding process through another aspect, information geometry. In paper A, the author shows an geometric approach using projections by three steps. Firstly, from the received symbols, a posterior probabilities (APP) of the transmit sequences are calculated, without taking account of the code structure. In this step, the complexity of APP is linear, because it only calculate the marginal distributions of the symbols, not the whole sequence. Secondly, we project this APP onto a linear family of code-compatible distributions. The third step is to maximize over all possible sequences.

Step One: Calculating the "Observed Distribution"

We try to calculate the a posterior pmf $p(\mathbf{s}|\mathbf{y})$, the observed distribution, where $\mathbf{s}$ is the symbol and $\mathbf{y}$ is the observation $\mathbf{y} = \mathbf{s} + \mathbf{n}, \mathbf{n} \sim \mathcal{N}(0, \sigma_n \mathbf{I})$. During this step, we ignore the code structure.

Since the channel is memoryless (each channel usage is independent), the pmf can be written as,

$$p(\mathbf{s}|\mathbf{y}) = \frac{p(\mathbf{s})}{p(\mathbf{y})} p(\mathbf{y}|\mathbf{s}) = \frac{p(\mathbf{s})}{p(\mathbf{y})} \prod_k p(y_k|s_k), \tag{3.1}$$

where $p(\mathbf{y})$ is a positive factor which does not depend on $\mathbf{s}$. If we assume the bits are uniformly distributed which results in uniformly distributed symbols, namely, $p(\mathbf{s})$ would be constant for all $\mathbf{s}$. Then the equation 3.1 can be reformed as

$$p(\mathbf{s}|\mathbf{y}) \propto \prod_k p(y_k|s_k). \tag{3.2}$$

As the correspondence between symbols s_k and subsequences of the interleaved bits $\tilde{\mathbf{x}}_k$, we can further write

$$p(\tilde{\mathbf{x}}|\mathbf{y}) = p(\mathbf{s}(\tilde{\mathbf{x}})|\mathbf{y}) \propto \prod_k p(y_k|\tilde{\mathbf{x}}_k) \tag{3.3}$$

Because the one-to-one mapping between the interleaved bits $\tilde{\mathbf{x}}$and the original bits $\mathbf{x}$, eventually we have the pmf

$$p(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \propto \prod_k p(y_k|\mathbf{x}_k) \tag{3.4}$$

where $\mathbf{x}_k = (x_{\pi^{-1}(kB)}, \ldots, x_{\pi^{-1}(k+1)B-1)})$. However, this $p(\mathbf{x})$ may also have non-zero value for sequences which are not valid codewords. So before maximization, we need to take the code structure into account.

Step Two: I-projection onto the Code Manifold

In this step, we should first build a manifold of distributions which has zero probability for sequences not belong to the codeword and all positive value for valid code words.

The manifold can be built as either exponential family or linear family.

Definition 3.1.1 Code-compatible distributions as an exponential family,

$$\mathscr{E}_C = \{p | p_e(\mathbf{x}) = 0 \forall \mathbf{x} \notin C\}. \tag{3.5}$$

Code-compatible distributions as a linear family,

$$\mathscr{L}_C = \{p | p_l(\mathbf{x}) : \mathbb{E}_p\{\mathbf{f}(\mathbf{x})\} = 0\} \tag{3.6}$$

Here C is the set of valid codewords. Equation 3.5 can be proved to be an exponential family, however, we will not use the result here. Let us just focus on code-compatible distributions as a linear family.

If we go to details about p_l, it can be written as

$$p_l(\mathbf{x}) = cq(\mathbf{x}) \exp(\sum_i \theta_i f_i(\mathbf{x})).$$

The function f_i stands for the parity-check equations of the code, e.g. $f_1 : x_1 \oplus x_3 \oplus x_5 = 0$. As $\exp(\theta_i f_i(\mathbf{x})) = I_i(\mathbf{x}) = \begin{cases} 0 & \text{if } f_i(\mathbf{x}) \neq 0 \\ 1 & \text{if } f_i(\mathbf{x}) = 0 \end{cases}$, when $f_i > 0$, θ_i has to be $-\infty$ to fulfill the requirements. Here $I_i(\mathbf{x})$ indicates the sequence $\mathbf{x}$ satisfy the i-th parity-check equation of the code. Eventually, the distribution p_l becomes

$$p_l(\mathbf{x}) = cq(\mathbf{x}) \prod_i I_i(\mathbf{x})$$

As described in Section 2.3 , when I-projection onto a factorisable linear family the result would be

$$p'(\mathbf{x}) = \frac{q(\mathbf{x})I_{\mathscr{C}}(\mathbf{x})}{\sum\limits_{\mathbf{x}\in\mathscr{X}} q(\mathbf{x})I_{\mathscr{C}}(\mathbf{x})}.$$

Here if let $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$ from the first step and $I_{\mathscr{C}}(\mathbf{x}) = \prod\limits_i I_i(\mathbf{x})$, the final result of step 2 is

$$p_{\mathscr{C}}(\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})I_{\mathscr{C}}(\mathbf{x})}{\sum\limits_{\mathbf{x}\in\mathscr{X}} p(\mathbf{x}|\mathbf{y})I_{\mathscr{C}}(\mathbf{x})}. \tag{3.7}$$

For all $p_{\mathscr{C}}(\mathbf{x})$ that $\sum\limits_{\mathbf{x}\in\mathscr{X}} p_{\mathscr{C}}(\mathbf{x}) = 1$.

Step Three: Maximization

In the third step, we perform the maximization over all valid probability distributions. It is easily shown that maximization of $p_{\mathscr{C}}$ is obtained in the projection step over all possible bit sequence leads to the same resulting code sequence estimate $\hat{c}$ as evaluating the MAP-criterion directly:

$$\begin{aligned}
\hat{c}_{\text{MAP}} &= \arg\max_{\mathbf{c}\in\mathscr{C}} p(\mathbf{c}|\mathbf{y}) && (3.8)\\
&= \arg\max_{\mathbf{c}\in\mathscr{C}} p(\mathbf{c}|\mathbf{y})I_{\mathscr{C}}(\mathbf{c}) && (3.9)\\
&= \arg\max_{\mathbf{c}\in\mathscr{C}} p_{\mathscr{C}}(\mathbf{c}) && (3.10)\\
&= \arg\max_{\mathbf{x}\in\mathscr{X}} p_{\mathscr{C}}(\mathbf{x}). && (3.11)
\end{aligned}$$

3.1.4 Rest Part of Paper A

After introducing the three-step procedure, Paper A has shown some practical algorithms and iterative decoding process to improve the three-step algorithms with many examples. Later, it introduces log-likelihood ratio (LLR) clipping to reduce the complexity of the decoding and the simulation results. The aim of LLR clipping is usually to avoid large computational

complexity by cutting away extreme values and limiting our attention to a smaller range. This technique is used commonly in iterative decoding and belief propagation fields. And this inspire us to use it in the second part of the thesis as one of the main technique to reduce the complexity in positioning algorithms. However, due to the limited length we will not show the details of LLR clipping here.

3.2 Paper B: Iterative Decoding as Dykstra's Algorithm with Alternate I-Projection and Reverse I-projection

This paper (paper B) is written by Florence Alberge, from Laboratoire des Signaux et Systemes, Univ. Paris-Sud. In this paper, the author has introduced a complete geometric interpretation of turbo-like iterative demodulation, particularly an interpretation on extrinsic propagation, resulting in floating manifolds to project which depend on the points that are to be projected. The author has also proved the iterative demodulation can be seen as the Dykstra's algorithm with I-projections and rI-projections.

3.2.1 Interpretation of Iterative Decoding

Demapping

Assume the exponential family $\mathscr{E}_{\mathscr{P}}$ as the product manifold, which is, the set of all distributions with independent components and bit length L_c. Let $p_{APP}(\mathbf{d})$ and $p_{APP}(\mathbf{c})$ be the a priori probabilities in the factorisable manifold $\mathscr{E}_{\mathscr{P}}$ with marginal distribution $p_{APP}(d_{km+i})$, $i = 1,\ldots,m; k = 1,\ldots L_c/m$ and respectively $p_{APP}(c_l)$, $l = 1,\ldots,L_c$, where $\mathbf{d}$ stands for interleaved coded bits and $\mathbf{c}$ for coded bits.

Based on the analysis before, we can have equation below,

$$p_{APP}(d_{km+i}=b)=p_{APP}(d_{km+i}=b|y_k)\propto\sum_{\mathbf{s}:s_k\in\Psi_b^i}p(\mathbf{y}|\mathbf{s})p(\mathbf{s}),\tag{3.12}$$

where $\mathbf{s}$ is the symbol sequence and $\Psi_b^i, b\in\{0,1\}$, denotes the subset Ψ which contains all symbols whose labels have the value b in the i-th position. Now, as described in Section 2.3, one can draw the conclusion that $p_{APP}(\mathbf{d})$ is the rI-projection of $p(\mathbf{y}|\mathbf{s})p(\mathbf{d};I)$ onto the factorisable manifold $\mathscr{E}_{\mathscr{P}}$. Here I and O in $p(\mathbf{d};I)$ and later $p(\mathbf{d};O)$ stands for the input and the output, respectively.

With an AWGN channel, $p(\mathbf{y}|\mathbf{s})$ is a Gaussian distribution,

$$p(\mathbf{y}|\mathbf{s})=a\exp(-\frac{||\mathbf{y}-\mathbf{s}||^2}{2\sigma_c^2}),$$

where a is the normalizer. Therefore, based on the analysis before, we can indicate that $p(\mathbf{y}|\mathbf{s})p(\mathbf{d};I)$ is the I-projection of $p(\mathbf{d};I)$ onto a linear family $\mathscr{L}_{\mathscr{M}}=\{p|\sum_{\mathbf{x}}p(\mathbf{x})=1,\sum_{\mathbf{x}}p(\mathbf{x})||\mathbf{y}-\mathbf{x}||^2=\alpha(\sigma_c^2)\}$.

Finally we can show the decoding and demodulation process into iterative projection process. Let $p_{APP}(\mathbf{d})=D_{\mathscr{M}}(p(\mathbf{d};I))$ where

$$D_{\mathscr{M}}:q\mapsto\overrightarrow{\pi}_f^{\mathscr{E}_{\mathscr{P}}}(\overleftarrow{\pi}_f^{\mathscr{L}_{\mathscr{M}}}(q)).\tag{3.13}$$

It is important that we should notice the linear family involved in the demapping is changing from one iteration to another [11]. Although it is changing, at each iteration, it must exist a linear family respectively where $p(\mathbf{y}|\mathbf{s})p(\mathbf{d};I)$ is the result of I-projection of $p(\mathbf{d};I)$.

Decoding

For the decoding part, there is a similar opposite process. First, we should have this APP,

$$p_{APP}(c_l = b) = a \sum_{\mathbf{c} \in \mathscr{R}_b^l} \mathbf{I}_{\mathscr{C}}(\mathbf{c}) \prod_{1 \le j \le L_c} p(c_j; I), \tag{3.14}$$

where a is the normalization factor, $\mathbf{I}_{\mathscr{C}}(\mathbf{c}) = \{1, 0\}$ denotes the indicator function of the code, and $\mathscr{R}_b^l$ presents the set of binary words of length L_c with value b in the l-th position.

From equation 3.14, one can conclude that $p_{APP}(\mathbf{c}) = D_{\mathscr{C}}(p(\mathbf{c}; I))$ where

$$D_{\mathscr{C}} : q \mapsto \overrightarrow{\pi}_f^{\mathscr{E}_{\mathscr{P}}}(\overleftarrow{\pi}_f^{\mathscr{L}_{\mathscr{C}}}(q)). \tag{3.15}$$

Conclusion

What has been shown above can be seen as APP propagation, since no extrinsic information is divided. This process can be interpreted as alternate projection algorithm of Csiszar [15], the convergence of which is built on closed convex sets. However, the exponential families are open sets with log-convex [13]. So we cannot guarantee this APP propagation of decoding and demapping converge.

In the next section, the author has shown that extrinsic propagation is a similar process as Dykstra' algorithm.

3.2.2 Dykstra's Algorithm with I/rI-projections

Dykstra's algorithm is an iterative procedure which asymptotically finds the nearest point of any given point onto the intersection of a family of closed convex sets [11]. It was first published by Dykstra in 1983 [16]. The difference from cyclic projections is that the Dykstra's algorithm does not use the result for the previous projection as the input of next projections directly, however, it modifies the outcome first and then projects again. There we can see

Algorithm 3.1 Dykstra's algorithm

1: Initialization

Let $s_{1,1} = r$ and let $p_{1,1} = \overset{\mathscr{C}_1\leftarrow}{\pi_f}(s_{1,1})$.

Let $s_{1,2} = p_{1,1} = r(p_{1,1}/s_{1,1})$, and set $p_{1,1} = \overset{\mathscr{C}_2\leftarrow}{\pi_f}(s_{1,2})$.

Let $s_{1,3} = p_{1,2}$ and set $p_{1,3} = \overset{\mathscr{C}_3\leftarrow}{\pi_f}(s_{1,3})$.

Let $s_{1,4} = p_{1,3}$ and set $p_{1,4} = \overset{\mathscr{C}_4\leftarrow}{\pi_f}(s_{1,4})$.

2: Iteration n

Let $s_{n,1} = p_{n-1,4}/(p_{n-1,1}/s_{n-1,1})$ and let $p_{n,1} = \overset{\mathscr{C}_1\leftarrow}{\pi_f}(s_{n,1})$.

For $i = 2$ to 4

Let $s_{n,i} = p_{n,i-1}/(p_{n-1,i}/s_{n-1,i})$ and let $p_{n,i} = \overset{\mathscr{C}_i\leftarrow}{\pi_f}(s_{n,i})$.

some similarity with extrinsic information which is achieved by the point-wise division from the outcome with the input of each block. In this paper B, we focus on the extended Dykstra's algorithm to I-projections in 1985 [16].

Let $\overset{\mathscr{C}_i\leftarrow}{\pi_f}$ denotes the I-projection onto $\mathscr{C}_i$, where $i \in \{1,2,3,4\}$. Note that the algorithm is valid for finite times of projections. In the process, all the products and divisions are point-wise operators: $\mathbf{u} = (\mathbf{pq})/\mathbf{r}$ means $\mathbf{u}(k) = (\mathbf{p}(k)\mathbf{q}(k))/\mathbf{r}(k)$, $\forall k$. Also, here the interleaver and deinterleaver is not taking into consider, since they have not influence on KL divergence.

One should notice that if $s_{1,1}(k) = 0$, then $p_{1,1}(k) = s_{1,1}(k) = 0$, in this condition we take $0/0 = 1$. The algorithm converges towards $u \in \mathscr{C} = \cap_1^4 \mathscr{C}_i$ and $D(u||r) = \arg\min_{p\in\mathscr{C}} D(p||r)$, namely, we try to fin the closest point to r in $\mathscr{C}$. However, classical cyclic projections do not necessarily converge to the closest pint in the intersection of the convex sets. Therefore, one can see that Dykstra's algorithm exhibits stronger properties than only alternating projections algorithms [11].

In the following, we compare the iterative decoding with Dykstra's algorithm.

It can be seen from the above two algorithms that they result the same in the projection onto $\mathscr{E}_{\mathscr{P}}$, in another words, they obtain the same sequences $\{p_{n,2}\}$ and $\{p_{n,4}\}$ which are the output of each sub-block in the iterative decoding, and they are intended to converge towards the same solution p^*. The author has proved a theorem about the two algorithms can behavior

Algorithm 3.2 Iterative decoding

1: Initialization

Let $v_{1,1} = (1/2^{Lc} \dots 1/2^{Lc})$ and let $p_{1,1} = \overset{\mathscr{L}_{\mathscr{M}}\leftarrow}{\pi}_f(v_{1,1})$.

Let $v_{1,2} = p_{1,1}$, and set $p_{1,2} = \overset{\rightarrow\mathscr{E}_{\mathscr{P}}}{\pi}_f(p_{1,1})$.

Let $v_{1,3} = p_{1,2}/v_{1,1}$ and set $p_{1,3} = \overset{\mathscr{L}_{\mathscr{C}}\leftarrow}{\pi}_f(v_{1,3})$.

Let $v_{1,4} = p_{1,3}$ and set $p_{1,4} = \overset{\rightarrow\mathscr{E}_{\mathscr{P}}}{\pi}_f(p_{1,3})$.

2: Iteration n

Let $v_{n,1} = p_{n-1,4}/v_{n-1,3}$ and let $p_{n,1} = \overset{\mathscr{L}_{\mathscr{M}}\leftarrow}{\pi}_f(v_{n,1})$.

Let $v_{n,2} = p_{n,1}$ and let $p_{n,2} = \overset{\rightarrow\mathscr{E}_{\mathscr{P}}}{\pi}_f(p_{n,1})$.

Let $v_{n,3} = p_{n,2}/v_{n,1}$ and let $p_{n,3} = \overset{\mathscr{L}_{\mathscr{C}}\leftarrow}{\pi}_f(v_{n,3})$.

Let $v_{n,4} = p_{n,3}$ and let $p_{n,4} = \overset{\rightarrow\mathscr{E}_{\mathscr{P}}}{\pi}_f(p_{n,3})$.

the same.

Theorem 3.2.1 Iterative decoding and Dykstra's algorithm with $r = (1/2^{Lc} \dots 1/2^{Lc})$, $\overset{\mathscr{C}_1\leftarrow}{\pi}_f = \overset{\mathscr{L}_{\mathscr{M}}\leftarrow}{\pi}_f$, $\overset{\mathscr{C}_3\leftarrow}{\pi}_f = \overset{\mathscr{L}_{\mathscr{C}}\leftarrow}{\pi}_f$, and $\overset{\mathscr{C}_2\leftarrow}{\pi}_f = \overset{\mathscr{C}_4\leftarrow}{\pi}_f = \overset{\rightarrow\mathscr{E}_{\mathscr{P}}}{\pi}_f$ lead to the same sequence of projected distribution $\{p_{n,2}\}$ and $\{p_{n,4}\}$.

It is very important for this achievements, however, there are compromise. Since original Dykstra's algorithm requires I-projections onto fixed closed convex sets, yet iterative decoding uses rI-projection onto log-convex sets. Therefore, the convergence results of Dykstra's algorithm cannot be applied directly. Particularly, there is no guarantee that the results converge to the closest point of r in the intersection of sets $\mathscr{E}_{\mathscr{P}}$, $\mathscr{L}_{\mathscr{C}}$ and $\mathscr{L}_{\mathscr{M}}$. But if we notice the duality between projections onto linear and exponential family, the iterative decoding can still be written with I-projections onto a varying linear family [17]. So the left difference with Dykstra's algorithm is limited to the non-fixed linear family.

3.3 Paper C: Belief Propagation, Dykstra Algorithm, and Iterated Information Projections

The main contribution of this paper (Paper C) is to rephrase the belief propagation algorithm as a hybrid of two projections algorithms: Dykstra's algorithm with cyclic Bregman projections and an alternating Bregman projection algorithm. Compare to paper B, paper C bring us further that it does not suffer from the variant sets which depend on the point to project. Meanwhile, paper C has extend our vision wider, not to limit ourselves with KL divergence but to use more general Bregman divergence which, as former described, has more diversity by changing function f.

Also paper C summarize the former works whose intention are to interpret belief propagation and iterative decoding as algorithms of projections. It serves well as an lib which gives clear history of the development of alternating projection algorithms and has proper comments for each work.

3.3.1 Bregman Projection with Legendre Type Function

As introduced in Section 2.2, suppose $f(\mathbf{x})$ is a strictly convex function and $\mathscr{C}$ is a convex subset of the domain $\mathscr{D}$, when $\mathbf{q}$ belongs to $\mathscr{D}$ yet not in $\mathscr{C}$, then the I-projection or left Bregman projection is the solution to the best approximation problem of $\mathbf{q}$ onto $\mathscr{C}$,

$$\overset{\mathscr{C}\leftarrow}{\boldsymbol{\pi}}_f(\mathbf{q}) = \arg\min_{\mathbf{p}\in\mathscr{C}} D_f(\mathbf{p},\mathbf{q}).$$

Furthermore, if f is the Legendre type, then the left Bregman projection $\overset{\mathscr{C}\leftarrow}{\boldsymbol{\pi}}_{f^*}(\boldsymbol{\theta})$, which is the conjugate function onto a convex $\mathscr{C} \subseteq \mathscr{D}^*$, can be transformed to be a certain right Bregman projection,

$$\overset{\rightarrow\hat{\mathscr{C}}}{\boldsymbol{\pi}}_f(\mathbf{q}) = \arg\min_{\mathbf{p}\in\hat{\mathscr{C}}} D_f(\mathbf{q},\mathbf{p}),$$

where $\hat{\mathscr{C}}$ defined as $\nabla f(\mathscr{C})$. Here it is very important to notice that we manage to do it by mapping through the coordinate change, demoed below,

$$\begin{aligned}
\nabla f(\overset{\mathscr{C}\leftarrow}{\pi}_{f^*}(\theta)) &= \nabla f(\arg\min_{\rho\in\mathscr{C}} D_{f^*}(\rho,\theta)) \\
&= \nabla f(\arg\min_{\rho\in\mathscr{C}} D_f(\mathbf{q},\nabla f(\rho))) \\
&= \arg\min_{\mathbf{p}\in\hat{\mathscr{C}}} D_f(\mathbf{q},\mathbf{p}) \\
&= \overset{\rightarrow\hat{\mathscr{C}}}{\pi}_f(\mathbf{q}).
\end{aligned}$$

If the function f here is negative entropy function, then left Bregman projection is identical to I-projection, respectively right Bregman projection as rI-projection.

3.3.2 Bregman Projection Algorithms

After revealing the "magical" changing the left projection to right projection by coordinate mapping, the author shows us some general projection algorithms. In the following procedure, we deal with two kinds of problems each in two different ways. By this comparison, it is clear to see the difference between alternating projections and Dykstra's algorithm.

Feasibility and Best Approximation

Let $\mathscr{C} = \bigcap_n \mathscr{C}_n$, $\mathbf{q}$ is a point outside $\mathscr{C}$, the best approximation is to find a point $\mathbf{p}$that minimize the divergence, namely, $\min_{\mathbf{p}\in\mathscr{C}} D_f(\mathbf{p},\mathbf{q})$. Suppose $\overset{(n)\leftarrow}{\pi}_f(\cdot)$ means, the left projections onto a set $\mathscr{C}_{[(n-1)\bmod N]+1}$, then, the solution to this problem by cyclic Bregman projections [18] is

$$\mathbf{p}_n = \overset{(n)\leftarrow}{\pi}_f(\mathbf{p}_{n-1})$$

where $\mathbf{p}_0 = \mathbf{q}$. It is not a complicated algorithm, however, it may not converge to the correct point. [16]

While on the other hand, Dysktra's algorithm may not be so intuitive to see, yet it in most cases works well. By starting with $\mathbf{p}_0 = \mathbf{q}$, $\mathbf{s}_j = 0$ where $j = \{0, -1, \ldots -(N-1)\}$, the Dykstra's algorithm can be reorganized in the way of Bregman projection in the following,

$$\mathbf{p}_n = \overset{(n)\leftarrow}{\pi}_f (\nabla f^*(\nabla f(\mathbf{p}_{n-1}) + \mathbf{s}_{n-N} \tag{3.16}$$

$$\mathbf{s}_n = \nabla f(\mathbf{p}_{n-1}) + \mathbf{s}_{n-N} - \nabla f(\mathbf{p}_n). \tag{3.17}$$

As the same with what we demoed in Section 3.2, equation 3.16 and 3.17 summarized Dykstra's algorithm, which results in a sequence $\{\mathbf{p}_n\}$ converge to the solution as $n \to \infty$.

Minimum Divergence in Different Sets

Let $\mathscr{C}_1$ and $\mathscr{C}_2$ both be a subset of $\mathscr{D}$, and has no intersection, $\mathscr{C}_1 \cap \mathscr{C}_2 = \emptyset$, the minimum divergence problem is to find a point in each subset, $\mathbf{p}' \in \mathscr{C}_1$, $\mathbf{q}' \in \mathscr{C}_2$, which gives the minimum divergence,

$$D_f(\mathbf{p}', \mathbf{q}') = \min_{\mathbf{p} \in \mathscr{C}_1, \mathbf{q} \in \mathscr{C}_2} D_f(\mathbf{p}, \mathbf{q}).$$

To find the solution with alternative projections, it is natural to arrival at the algorithm [15] [19] below,

$$\mathbf{p}_n = \overset{\mathscr{C}_1\leftarrow}{\pi}_f (\mathbf{q}_{n-1}) \tag{3.18}$$

$$\mathbf{q}_n = \overset{\rightarrow\mathscr{C}_2}{\pi}_f (\mathbf{p}_n). \tag{3.19}$$

Since the Dykstra's algorithm with cyclic Bregman projection uses only left projections, it does not fit to solve this kind of problems. However, the author gives us an special case which by function half the squared Euclidean distance, $f(\cdot) = \frac{1}{2}||\cdot||^2$. In this situation, the left projection is equal to right projection. Thus we can apply Dykstra's algorithm with cyclic

Bregman projection to this problem and procedure is:

$$\mathbf{p}_n = \overset{\mathscr{C}_1}{\pi}_f(\mathbf{q}_{n-1} + \mathbf{v}_{n-1})$$

$$\mathbf{v}_n = \mathbf{q}_{n-1} + \mathbf{v}_{n-1} - \mathbf{p}_n$$

$$\mathbf{q}_n = \overset{\mathscr{C}_2}{\pi}_f(\mathbf{p}_{n-1} + \mathbf{w}_{n-1})$$

$$\mathbf{w}_n = \mathbf{p}_n + \mathbf{w}_{n-1} - \mathbf{q}_n.$$

The result of the algorithm is shown to converge to the minimum divergence [20].

3.3.3 Belief Propagation Interpreted as a Modified Dykstra's Bregman Projection Algorithm

Later in Paper C, as one of the major results, the author rephrase belief propagation as the modified dykstra's algorithm with Bregman projections. However, it is very hard to explain the whole process within this limited space here. So we would only present the theorem here and refer the reader who are interested in prove process back to the original paper.

Let the two manifold on which we perform the projections be $\mathscr{P}$ and $\mathscr{Q}$, defined below:

$$\mathscr{P} = \{\theta : \nabla f^*(\theta) = q(\mathbf{x}^1, \ldots, \mathbf{x}^K) = \prod_{k=1}^{K}\prod_{i=1}^{M} q_{k,i}(x_i^k)\} \tag{3.20}$$

$$\overset{\wedge}{\mathscr{P}} = \nabla f(\mathscr{P}) \tag{3.21}$$

$$\mathscr{Q} = \{r \in \mathscr{D} : r(\mathbf{x}^1, \ldots, \mathbf{x}^K) = 0 \, if \, \mathbf{x}^i \neq \mathbf{x}^j \forall i \neq j\} \tag{3.22}$$

$$[\mathbf{z}_{-1}]_i = \prod_{k=1}^{K} g_k(\mathbf{b}_i^k) \tag{3.23}$$

where $i = 1, 2, \ldots M$ is the ordinal of the variable nodes and $k = 1, 2, \ldots K$ is the ordinal of the factor nodes for the factor graphs which belief propagation use, g_k is a factor of the

likelihood function $p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{K} g_k(\mathbf{x})$ and $\mathbf{b}_i = [\mathbf{b}_i^1, \ldots, \mathbf{b}_i^K]$ is equal to the integer i with the KM bit binary representation.

Theorem 3.3.1 The belief propagation algorithm can be reformed as a process of the following Bregman projection algorithm

$$\begin{aligned}
\mathbf{k}_n &= \overset{\wedge}{\underset{f}{\pi}}{}^{\mathscr{P}}_{\rightarrow}(\nabla f^*(\nabla f(\mathbf{z}_{n-1}) + \boldsymbol{\sigma}_{n-1})) && (3.24)\\
\boldsymbol{\sigma}_n &= \nabla f(\mathbf{z}_{n-1}) + \boldsymbol{\sigma}_{n-1} - \nabla f(\mathbf{k}_n) && (3.25)\\
\mathbf{r}_n &= \overset{\mathscr{Q}}{\underset{f}{\pi}}{}_{\leftarrow}(\nabla f^*(\nabla f(\mathbf{k}_n) + \boldsymbol{\tau}_{n-1})) && (3.26)\\
\mathbf{z}_n &= \overset{\wedge}{\underset{f}{\pi}}{}^{\mathscr{P}}_{\rightarrow}(\mathbf{r}_n) && (3.27)\\
\boldsymbol{\tau}_n &= \nabla f(\mathbf{k}_n) + \boldsymbol{\tau}_{n-1} - \nabla f(\mathbf{z}_n) && (3.28)
\end{aligned}$$

for $n \geq 0$, f is the negative Shannon entropy, and the initialization is$\boldsymbol{\sigma}_{-1} = \boldsymbol{\tau}_{-1} = \mathbf{0}$ with $\mathbf{z}_{-1}$ in 3.23.

3.4 Paper D: Stochastic Reasoning, Free Energy and Information Geometry

In this paper (paper D), the author offers a special view through which we can compare the network of factor graphs (factor notes and edges) and the elements in information geometry (points, lines and faces). With this unique angle, it may help us gain deeper understanding of both belief propagation and information geometry.

For a given distribution $q(\mathbf{x})$, it is possible to reform the expression in log domain as a polynomial of $\mathbf{x}$ up to degree n, since every x_i is binary,

$$\ln q(\mathbf{x}) = \mathbf{h} \cdot \mathbf{x} + \sum_{r=1}^{L} c_r(\mathbf{x}) - \psi_q$$

where $\mathbf{h}\cdot\mathbf{x}=\sum_i h_i x_i$ as the linear term, $c_r(\mathbf{x})$, $r=1,\ldots,L$, is a simple polynomial representing the r-th clique connecting relative variables, and ψ_q is the normalizer.

The definition of $c_r(\mathbf{x})$ is clearer if we view it as a Boltzmann machine, $c_r(\mathbf{x})=w_{ij}^r x_i x_j$, it forms an edge in the undirected graphs, here r is the index of mutual linking between two variables x_i and x_j.

Given a distribution, we can build a set of distributions around the distribution characterized by parameter θ and v,

$$S=\{p(\mathbf{x};\theta,v)|p(\mathbf{x};\theta,v)=\exp(\theta\cdot\mathbf{x}+v\cdot c(\mathbf{x})-\psi(\theta,v))\}$$

where $v\cdot c(\mathbf{x})=\sum_{r=1}^{L} v_r c_r(\mathbf{x})$. When $\theta=\mathbf{h}$ and $v=1_L$, $q(\mathbf{x})=p(\mathbf{x};\theta,v)$.

One special subset should be noticed that

$$M_0=\{p(\mathbf{x};\theta,v)=\exp(\mathbf{h}\cdot\mathbf{x}+\theta\cdot\mathbf{x}-\psi(\theta))|\theta\in\mathbb{R}^n\}.$$

In this subset, one can see no mutual interactions between any two variables, so it is the product of marginal distributions of the original distribution $q(\mathbf{x})$, i.e. $\prod_{i=1}^{n} q(x_i)\in M_0$. As the subset M_0 has the coordinate θ, if one is interested in deriving marginal distributions, then, it is important to find the corresponding θ in M_0 or $p_0(\mathbf{x};\theta)$. Later the author has shown the maximization of the posterior marginals (MPM) is equal to rI-projection of $q(\mathbf{x})$ onto M_0. This conclusion is very important in interpreting belief propagation in information geometry.

Before introducing the belief propagation, the author shows three kinds of distributions, the marginal distributions, the original distribution and one of the key distribution for the following process, which has only one edge and can be defined as,

$$p_r(\mathbf{x};\zeta_r)=\exp(\mathbf{h}\cdot\mathbf{x}+c_r(\mathbf{x})+\zeta_r\cdot\mathbf{x}-\psi(\theta)),\ r=1,\ldots,L.$$

Algorithm 3.3 Belief Propagation

1: Initialization
Let $t = 0$, $\zeta_r^t = 0$, $r = 1, \ldots, L$
2: Iteration $t = t + 1$
Let $\xi_r^t = \overrightarrow{\pi}_f^{M_0}(p_r(\mathbf{x}; \zeta_r^t)) - \zeta_r^t$ and
$\zeta_r^{t+1} = \sum_{r'=r} \xi_{r'}^t$
$\theta^{t+1} = \sum_r \xi_r^t = \frac{1}{L-1} \sum_r \zeta_r^{t+1}$

Let $M_r = \{p_r(\mathbf{x}; \zeta_r) | \, \zeta_r \in \mathbb{R}^n\}$ denotes the exponential family of distribution $p_r(\mathbf{x}; \zeta_r)$ whose coordinates is ζ_r. In the subset M_r, only r-th edge is considered and all other edges are replaced by the linear term $\zeta_r \cdot \mathbf{x}$.

In belief propagation, firstly, it updates the coordinates ζ_r of $p_r(\mathbf{x}; \zeta_r)$ using the integrated information θ, then all the information from $p_r(\mathbf{x}; \zeta_r)$, $r = 1, \ldots, L$ merges to marginal distributions $p_0(\mathbf{x}; \theta)$, and the process continues iteratively. So the belief propagation can be summarized as the following algorithm.

Let $\{\xi_r^*\}$, $\{\zeta_r^*\}$ and θ^* denotes the converged point of belief propagation. Then the author reveals another theorem,

Theorem 3.4.1 The converge of belief propagation $\{\zeta_r^*\}$ and θ^* satisfies

1) m-condition: $\theta^* = \overrightarrow{\pi}_f^{M_0}(p_r(\mathbf{x}; \zeta_r^*))$

2) e-condition: $\theta^* = \frac{1}{L-1} \sum_r \zeta_r^*$.

Also we can interpret the two condition in geometrical view. As one may noticed, the converge is characterize by two parameters $\{\zeta_r^*\}$ and θ^*, it is natural to think the variation of these two parameters can result in two submanifolds,

$$M^* = \{p(\mathbf{x}) | \sum_{\mathbf{x}} p(\mathbf{x})\mathbf{x} = \sum_{\mathbf{x}} p_0(\mathbf{x}; \theta^*)\mathbf{x}\}$$

$$E^* = \{p(\mathbf{x}) | \, p(\mathbf{x}) = c \cdot p_0(\mathbf{x}; \theta^*)^{t_0} \prod_{r=1}^{L} p_r(\mathbf{x}; \zeta_r^*)^{t_r}, \sum_{r=1}^{L} t_r = 1\}$$

where c is the normalization factor. The m-condition means the submanifold M^* includes all the points $p_r(\mathbf{x};\zeta_r^*)$ and the marginal distribution $p_0(\mathbf{x};\theta^*)$, meanwhile, M^* is orthogonal to all the subsets M_r and M_0, i.e., they are the rI-projection to each other. On the contrary, e-condition suggests the submanifold E^* contains all $p_r(\mathbf{x};\zeta_r^*)$, $p_0(\mathbf{x};\theta^*)$ and the original distribution $q(\mathbf{x})$.

One should notice that once $q(\mathbf{x}) \in M^*$, then the $p_0(\mathbf{x};\theta^*)$ is the true marginalization where the belief propagation converges. However, this may not happen if the factor graphs has circles. But if it is a tree structure graph, one can prove that $q(\mathbf{x}) \in M^*$ always holds.

Chapter 4

Comparison and Conclusions

In this chapter, we offer the comparison between these four papers. However, all papers are great helpful during the study of information geometry and its interpretation of iterative decoding/demapping and belief propagation. Here we try list each paper with its own distinguishing features and special area which it focuses on. Hope this can provide somehow a big picture to help the readers understand the content much easier and find interests much quicker.

4.1 Contributions in Each Paper

The paper A, Information Geometric Analysis of Iterative Receiver Structures, is a master thesis written by Stefan Schwandter, in 2008. The thesis first introduces some basic concept of information geometry, later it provides a alternative projection algorithm which interprets the decoding process. However, the interpretation is not fully complete, since it lacks the iterative procedure, i.e. after two projections it does maximization and no iterative projections. Without taking care of extrinsic information, i.e. no geometrical interpretation of dividing self-information, the projection algorithm here is similar to what introduced in paper B, the APP propagation. So one can consider paper A as a foundational stone to step into the in-

formation geometry and iterative decoding. Furthermore, as a thesis which has less pages limitation than papers, it provides a bunch of detail examples, which are extremely helpful to understand and visualize the problems in the beginning. Another contribution is that the paper A has a great concern about the decoding complexity. Bearing this goal, the author has invited log-likelihood ratio (LLR) clipping algorithm which prove to be effective to reduce the computational power.

The paper B bring us much further by two major contributions. The first is the complete APP propagation. Although in this procedure the subset to be projected onto is variate and depending on the point to be project, the two alternative projections of APP propagation make nice approach to the real "iterative" process. As we know, this process is not necessarily converge to the optimum solution, the author provides the second research result, linking Dykstra's algorithm with I/rI-projections and iterative decoding. In the paper, it has shown the similarity of these two algorithm and a theorem to prove that it leads to the same sequence of distributions. In this algorithm, the extrinsic information is obtained, therefore, with stronger properties, the Dykstra's algorithm with I/rI-projection can result in better solutions.

Standing on many former studies, the paper C has a huge success by proving the belief propagation as an instance of a hybrid of two projection algorithms. Compare to paper B, it does not suffer from the variable subset to project on. Besides this extraordinary contribution, the paper C has also extend our vision to not only use KL divergence but more general metric, the Bregman divergence, which includes many other metric by changing the function f. This extra degree of freedom induces another meaningful achievement, the convergence of belief propagation happens when using symmetric Bregman divergence. It has also systematized coverage and summary about the former works both in information geometry and convex projection algorithms, which can be employed as a good library or road map if one wants to dig deep into this field. With many advantages, the paper C still has some shortcomings. Without detailed examples, it would be hard for someone who is new to this problem. Even

Table 4.1: Short comparison between 4 papers, contributions

Papers	Contributions
Paper A	Detailed examples, easy to hand on
Paper B	Big step towards fully interpretation of BP
Paper C	Huge success, complete interpretation of BP, state of art
Paper D	Visualized details, focusing on convergence condition

Table 4.2: Short comparison between 4 papers, shortcomings

Papers	Shortcomings
Paper A	Not good at the interpretation of the whole iterative process
Paper B	The manifold is floating, depending on the point to project
Paper C	Hard to understand, high requirements of math skills
Paper D	The interpretation is not fully complete comparing to B and C

if there may be some flaws, this work is remarkable.

After the first three papers, the reason to present the paper D is because it provides a unique angle for us to see how the structures of belief propagation, in particular factor graphs, looks like under the information geometrical interpretation. With the visualized lines and faces, instead of the factors and edges, it offers a brand new way for the well-known algorithm and may be inspiring for better solutions. In addition, the work in paper D gives a clear image about when the belief propagation can converge and has the true marginalization.

4.2 Conclusion and Future Works

This part of thesis first introduces the basic knowledge of information geometry, which are important submanifolds, the distance metric, KL divergence as an instance of Bregman divergence, and two common projections which forms the basic procedure of iterative projections. Followed by the core work of this thesis, introducing the four representative papers. Here we only focus on the major relevant contributions of each paper. As the limitation of the length of the thesis, we cannot fully cover many important proofs and details which may helpful to understand. However, the aim of the thesis would be to summarize the states of art in the field

of information geometrical interpretation of iterative decoding/demapping and belief propagation. Also we attempt to offer an easier way and an all-around view to understand this area for more people in the communication community. In the last chapter of part I of the thesis, we list some comments on each paper about their strong points and short comings. However, all these paper we chooses are all magnificent and remarkable which provide boundless assistance during the research.

The future work may provide more complete analysis of belief propagation in information geometry, and explore more about the condition of convergence for factor graphs with loops, and maybe with the performance boundaries from the tree-structure graphs. Also it would be interesting to apply the theoretical algorithms to the real decoders and verifying whether the structure of factorization or the manifolds can induce in better results. How to reduce the complexity of the decoders may also lie within the investigation in information geometrical interpretation of belief propagation.

Part II

Geometrical Structure-aware Bayesian Positioning Algorithm with Dynamic Grid and Clipping

Chapter 5

Introduction

The rapid development of location-aware technologies has brought us to a new era which has revolutionized many aspects of commercial, public service, and military usage with the application of high accurate ubiquitous location awareness [21,22]. Besides the huge success from the explosion of GPS-based techniques, many positioning algorithms have been proposed to achieve location in the GPS-denied environment where the visibility of GPS satellites is limited, e.g., indoor scenarios [23–25]. During the last few years, self-positioning by the network itself, especially wireless sensor networks, has received growing interest [26–28].

Commonly, a positioning algorithm aims to locate the nodes with unknown position (referred to as agents) from the position of reference nodes (referred to as anchor) and some measurements between agents and anchors. There are many mature techniques for measurement purpose, e.g., time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS) [29–32]. In this paper, we mainly focus on measurements which give the distance estimated between each node. Given a model and statistics of measurement errors, we can introduce the classical estimation algorithms to solve the problem, such as maximum likelihood (ML) [33, 34], nonlinear least squares [35], and linear least squares estimators [36–38].

Compared to the classical estimators, the Bayesian positioning algorithm known as sum-product algorithm over a wireless network (SPAWN), has outstanding performance in many network scenarios [39]. Information shared between agents includes spatial distributions over the whole map instead of only estimated positions. This additional message-passing and information-fusing scheme guarantees SPAWN with a good performance. However, because the spatial distributions as messages are based on grid over the map, the size of the messages is seriously effected by the geometrical size and the resolution of the map. So this grid-based approach of SPAWN suffers from high computational complexity when fusing the messages from neighbor agents. For the same reason, the performance is limited by the resolution. Also when it comes to practical usage, large packets resulted by complete distributions are hard to be send through networks.

In this thesis, we proposed three major techniques to reduce the drawbacks of SPAWN while maintaining the outstanding performance. Clipping is the most effective way to reduce complexity by using a small number of points with the highest probability instead the complete distribution. The geometrical pre-location can save the trouble to initialize the whole map, but to only focus on where the agent mostly can be. The dynamic grid controls the boundary size and resolution of blocks. With these three methods, we achieved contributions below:

- Reduce the complexity completely, 3 degree of freedom in parameter design for different network scenarios.
- Increase the resolution and enable the algorithm effectively handle arbitrary large maps.
- Largely reduce the total amount of communication between agents, namely the network traffic, and total iteration times needed to convergence by only allowing agents with a certain estimation broadcast.
- Largely reduce the size of the packet as the message broadcasting to neighbor agents.

- Increase the reliability of the estimates and reduce the chance of broadcasting fault estimation by awareness of network geometrical structure.

This thesis is organized as follows. In chapter , the system model is provided, together with a brief description of SPAWN algorithm. Chapter presents the new algorithm in detail. The network scenarios and numerical results are shown in chapter. Finally, we draw our conclusions and discuss the area for future research in chapter.

Chapter 6

System Model and Brief Description of SPAWN Algorithm

6.1 System Model

We assume two types of nodes in a wireless network: agent , with unknown position $\mathbf{x}_i \in \mathbb{R}^2$; anchor, as the reference node with position $\mathbf{a}_i \in \mathbb{R}^2$, i is the index of the agent and anchor. Each agent estimates the distance to every anchor in its communication range, results in $\hat{d}_{i-j} = \left\|\mathbf{x}_i - \mathbf{a}_j\right\| + n_{i-j}$, where n_{i-j} is the ranging noise with distribution $n_{i-j} \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.1m$. The set of anchors that agent i can communicate with is denoted by $\mathbf{S}_i^d$. Anchors which agent i cannot communicate directly, yet belongs to $\mathbf{S}_j^d$, the set of its neighbor agent j , are collected into set $\mathbf{S}_i^{ind}$. In our new algorithm, we assign agents with different cardinality of sets into 4 classes of conditions (an example presented in Figure6.1).

- Condition 3:

$$|\ \mathbf{S}_i^d\ | \geqq 3,\ cond_i = 3$$

- Condition 2:

$$| \mathbf{S}_i^d | = 2, \; | \mathbf{S}_i^d \cup \mathbf{S}_i^{ind} | \geqq 3, \; cond_i = 2$$

The set $\mathbf{S}_i^{ag}$ contains the neighbor agents of agent i who has reliable estimation of its own position after iterations. This is described with more detail at Section III and IV.

- Condition 1:

$$| \mathbf{S}_i^d | = 2, \; | \mathbf{S}_i^d \cup \mathbf{S}_i^{ind} | = 2, \; | \mathbf{S}_i^{ag} | \geqq 1$$

or

$$| \mathbf{S}_i^d | \leqq 1, \; | \mathbf{S}_i^d | + | \mathbf{S}_i^{ag} | \geqq 3, \; cond_i = 1$$

- Condition 0:

$$| \mathbf{S}_i^d | + | \mathbf{S}_i^{ag} | < 3, \; cond_i = 0$$

6.2 Brief Description of SPAWN

Sum-product algorithm is an effective algorithm to compute marginal distributions on factor graphs with messages passing between nodes. The SPAWN algorithm develops a factor graph according to the network topology, passes messages on the factor graph and runs sum-product algorithm to compute the distributions. To begin the SPAWN algorithm. each agent is assigned with uniform distribution over the entire map. Firstly, the agent updates its priori distribution with the distance estimates and positions of the anchors available. After self estimation, these distributions, denoted by $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$ also known as belief, are sent to neighbor agents as messages. The messages receiver fuses the information to form a new distribution, $b_{\mathbf{x}_i}^{(1)}(\mathbf{x}_i)$ and broadcast the updated distribution again. This process is parallel executed on every agent and iterate a few times, until either the belief converges or forced to stop. The algorithm is summarized in Algorithm 1.

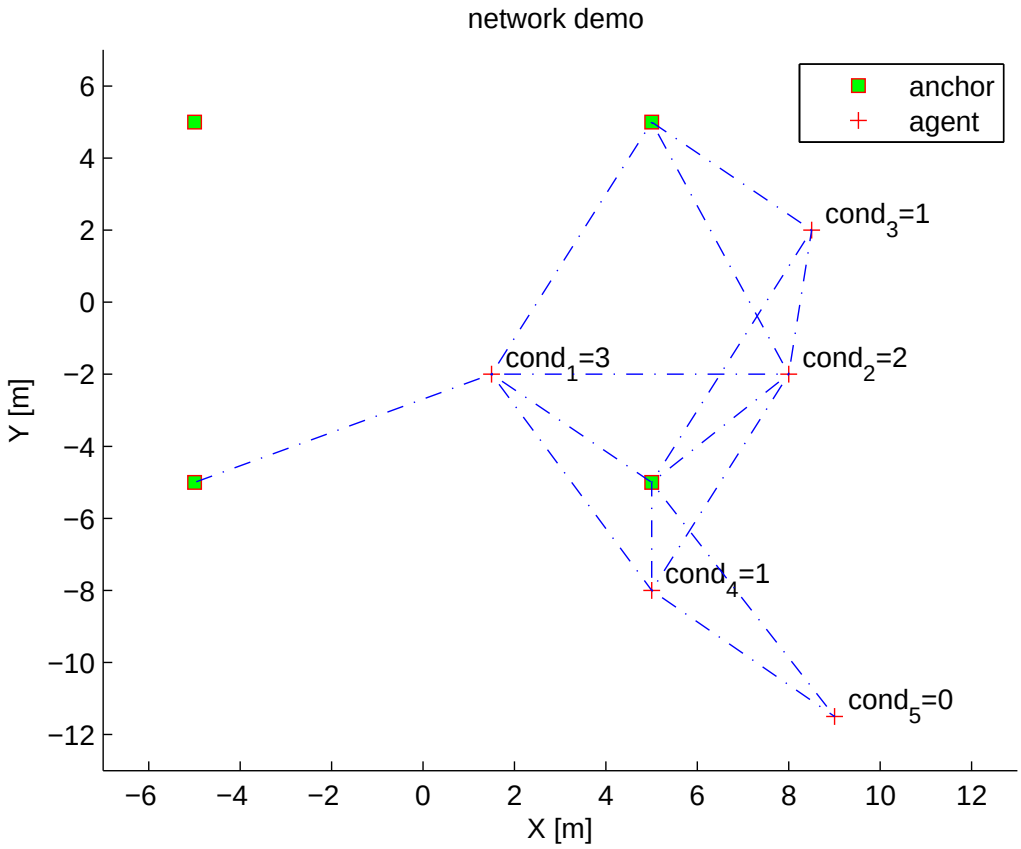

Figure 6.1: A demonstration of agent condition

Algorithm 6.1 Original SPAWN

1: Initialize $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$, $\forall i$, over $\mathbf{L}$, and broadcast.

$$b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i) \propto \sum_j p(\hat{d}_{i-j}|\mathbf{a}_j, \mathbf{x}_i),\ j \in \mathbf{S}_i^d$$

2: **for** $t = 1$ to T **do** {iteration index}
3: **for** agent $i = 1$ to I **do** {agent index}
4: receive all messages from neighbor agents $b_{\mathbf{x}_j}^{(t-1)}(\cdot)$
5: create updating distributions over $\mathbf{L}$ by messages ,

$$m_{j-i}(\mathbf{x}_i) \propto \int p(\hat{d_{j-i}} \mid \mathbf{x}_i, \mathbf{x}_j) b_{\mathbf{x}_j}^{(t-1)}(\mathbf{x}_j) d\mathbf{x}_j$$

6: update new belief, and broadcast

$$b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i) \propto b_{\mathbf{x}_i}^{(t-1)}(\mathbf{x}_i) \prod_j m_{j-i}(\mathbf{x}_i)$$

7: **end for**
8: **end for**

By analyzing the process of SPAWN, it is clear to see the shortcomings. Firstly, the belief from neighbor j contains distributions over the entire map $\mathbf{L}$, where $\mathbf{L} = l^2$, l denotes the boundary length of the map. Let r denotes the resolution of the map, the complexity of messages fusing at receiver is roughly related to $(\frac{l}{r})^4$. So it is unpractical to fuse this information which results in large complexity. Secondly, if l is large or r is fine, even initializing $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$ and sending the messages will be unaffordable. In Section III, we propose a new algorithm to avoid the drawbacks above happen.

Chapter 7

Geometrical Structure-aware, Dynamic Grid, and Clipping

7.1 General Ideas

Based on the study of existing range-oriented positioning algorithms, we find the basic requirements of reliable estimation on 2 dimension cases are:

- Requirement 1: At least 3 reliable sources where its self position is accurate.
- Requirement 2: The geometrical structure of sources is in low-risk situation.

Requirement 1 is easy to conceive, yet by this basic idea, we can design an algorithm with more effective cooperation scheme, which gives more priority to agents with enough reliable sources (either anchors or agents with good estimations).

In requirement 2, by low-risk situation , we mean if the source nodes are placed well, not too close to each other or line up, in which situation it has more probability to create ambiguity for the receiver even the number of source nodes is greater than 3. Traditional positioning algorithms is easily blind to see the geometrical structure of source nodes, so we propose a

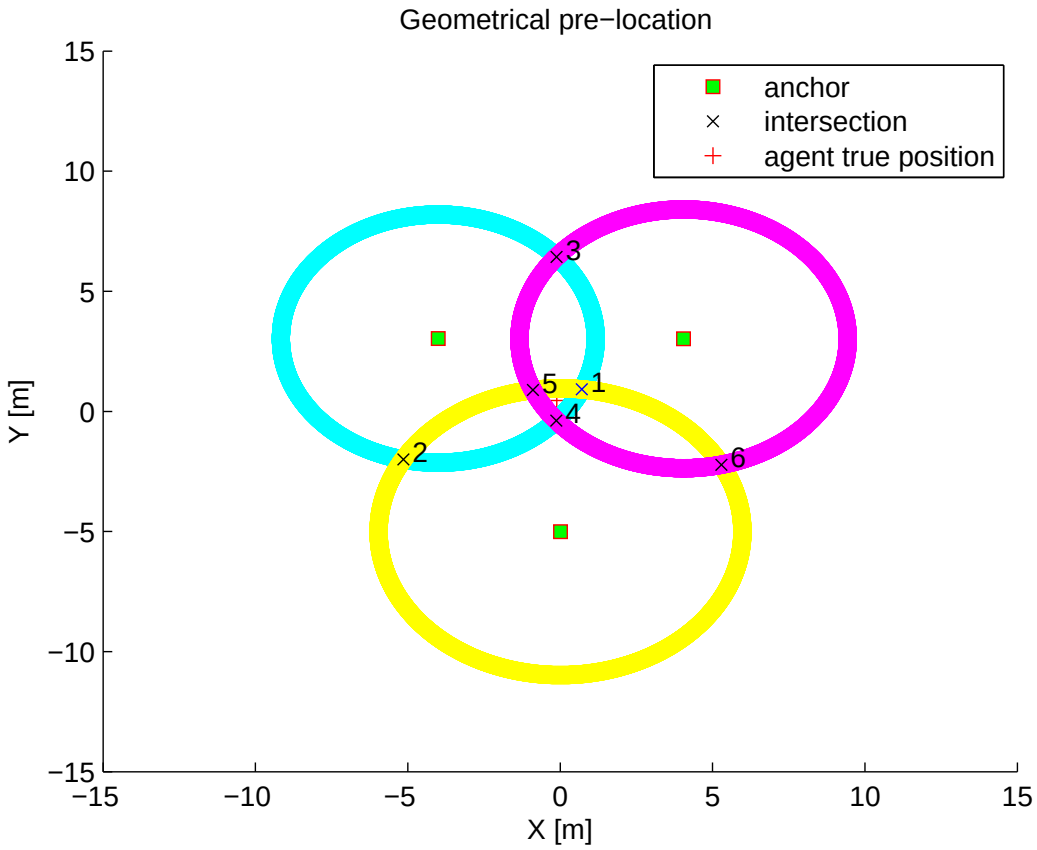

Figure 7.1: Geometrical pre-location

method, named geometrical pre-location, which is a similar process like triangulation location. However, with ranging noise, the intersections of circles will not converge to the true position, see in Figure 7.1. Yet geometrical pre-location is good enough to point out a block of area where the true position most likely to be and give warning information if the sources structure is with risk.

7.2 Clipping

Clipping method recently attracts some attention of soft decoding community, mainly used to clip log-likelihood ratios (LLRs) [40]. It provides a promising performance-complexity

tradeoff. In some works related to information geometry, clipping is used to generate clipping manifolds, which also serves complexity reduction [41]. Since iterative decoding process is similar to belief propagation in cooperative positioning, it enlightens us for applying it to SPAWN.

The essential idea of clipping is to replace the belief with a unique value once it goes beyond a certain threshold. Here we abandon large part of distribution which has little effects on updating the belief. This can be formulated as follow:

The clipped belief $c_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i)$ with clipping threshold α from original belief $b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i)$ is denoted by $c_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i) = clip(b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i),\ \alpha)$, where

$$c_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i) = b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i),\ \forall \mathbf{x}_i \in \varphi_i,$$

$$\varphi_i = \left\{ \mathbf{x}_i \mid b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i) \geqq Max(b_{\mathbf{x}_i}^{(t)}(\mathbf{x}_i)) - \alpha \right\}$$

7.3 Algorithm Proposed

Based on the general ideas above, we propose the algorithm with more effective cooperation schemes, low complexity, and less network traffic requirements (Shown in Algorithm 7.1).

The algorithm starts with processing the agents in Condition 3: agent i, $cond_i = 3$. Instead of initializing $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{L}$, we apply geometrical pre-location to limit our attention on a block with size $\mathbf{L}_3^i = l_3^2$, covering area of Intersection 1, Intersection 4, and Intersection 5, in Figure 7.1. This will avoid wasting computational power on some uninterested area. After estimating $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{L}_3^i$, clipping is used to select a few points on gird $\mathbf{L}_3^i$ with belief value above a clipping threshold α_3. The clipped belief $c_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$, as message, is broadcast to all neighbor agents also in condtion3, agent j, $cond_j = 3$. After fusing the received messages and clipping again, the estimation is improved and more robust. This new belief $c_{\mathbf{x}_i}^{(1)}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{L}_3^i$ is broadcaster as a reliable source.

Algorithm 7.1 Proposed Algorithm

for $n = 3,\ 2,\ 1,\ and\ 0$ **do** {condition index}
  for agent $i = 1$ to I_n, $cond_i = n$ **do** {agent index}
    Initial $b^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i),\ \forall i$, over $\mathbf{L}^i_n$,

$$b^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i) \propto \sum_j p(\hat{d}_{i-j}|\mathbf{a}_j, \mathbf{x}_i),\ a_j \in \mathbf{S}^d_i$$

    clip belief with threshold α_n,

$$c^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i) = clip(b^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i),\ \alpha_n)$$

    If $n = 3,\ 2,\ or\ 0$ broadcast $c^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i)$ to neighbor agent j, $cond_j = n$
    create update distributions over $\mathbf{L}^i_n$ by received messages ,

$$m_{j-i}(\mathbf{x}_i) \propto \int p(\hat{d_{j-i}} \mid \mathbf{x}_i, \mathbf{x}_j) c^{(0\,or\,1)}_{\mathbf{x}_j}(\mathbf{x}_j) d\mathbf{x}_j,$$

$$\forall \mathbf{x}_i \in \varphi_i\ \forall \mathbf{x}_j \in \varphi_j$$

    update new belief,

$$b^{(1)}_{\mathbf{x}_i}(\mathbf{x}_i) \propto b^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i) \prod_j m_{j-i}(\mathbf{x}_i),\ \forall \mathbf{x}_i \in \varphi_i,$$

    clip again and broadcast to all neighbor agents

$$c^{(1)}_{\mathbf{x}_i}(\mathbf{x}_i) = clip(b^{(1)}_{\mathbf{x}_i}(\mathbf{x}_i),\ \alpha_n)$$

  end for
end for

Second step is to execute the agents in Condition 2: agent i, $cond_i = 2$. Since agents in Condition 2 has information from two anchors, the geometrical pre-location chooses two blocks to cover the two intersection respectively and initialize $b^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{L}_2^{i1}$ and $\mathbf{x}_i \in \mathbf{L}_2^{i2}$, $\mathbf{L}_2^i = \mathbf{L}_2^{i1} \cup \mathbf{L}_2^{i2}$, where the block length of $\mathbf{L}_2^{i1}$ and $\mathbf{L}_2^{i2}$is l_2. The rest procedure is similar as Condition 3, by messages from neighbor agents in Condition 3 or Condition 2, we can peak a unique block with good estimation and clipped again which finally give us $c^{(1)}_{\mathbf{x}_i}(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbf{L}_2^{i1}$ or $\mathbf{x}_i \in \mathbf{L}_2^{i2}$.

The next is to locate the agents in Condition 1: agent i, $cond_i = 1$. No matter how many anchors available, there are at least three reliable reference nodes for an agent in Condition 1. This is enough to use geometrical pre-location to create a block $\mathbf{L}_1^i = l_1^2$. Compare to $\mathbf{L}_3$ and $\mathbf{L}_2$, $l_1$is designed larger for the safety reason. Another difference is $c^{(0)}_{\mathbf{x}_i}(\mathbf{x}_i)$ is not broadcast in this step, since it is less powerful than $c^{(1)}_{\mathbf{x}_i}(\mathbf{x}_i)$ and without crucial information. Since after each iteration the agents in Condition 1 have reliable estimation and this helps more agents become Condition 1, this step will continuously iterate several times until no more agents in the network qualify Condition 1.

Agents in Condition 0 have the last priority. For the agents with two reliable nodes, follows similar process of agents in Condition 2. Hopefully, its neighbor agents which also in Condition 0 have at least one different reference nodes, in this situation the cooperation will locate both agents. Otherwise, without enough information, the agent cannot locate itself accurately. In this case, we use the position of the anchor, if any, as its final estimation.

Compared to original SPAWN, the new algorithm is less resistible of error, the fault information will propagate and cause serious damage to location accuracy due to "one-way" broadcast and less information in the messages. So in geometrical pre-location, we examine the geometrical structure of the reference nodes very carefully. By the distance between each nodes and whether the nodes are placed line-like, we can distinguish whether it is in low-risk scenarios . If the high-risk structures happen, the algorithm will band the agent to estimate

and broadcast the belief until enough reference nodes in good structure available. If after iterations, still no the resource satisfy the requirements, the agent will estimate its position follow the normal procedures, yet will not send high-risk messages to neighbors.

Chapter 8

Numerical Results and Complexity Analysis

8.1 Simulation Scenarios

We generate 40 network scenarios with map size $100m \times 100m$, 13 manually placed anchors, and 100 random placed agents in each scenarios. The communication range of each node is 20 m and the standard deviation of the ranging noise,σ, is 0.1 m. For the newly proposed algorithm, we consider both location performance and complexity costs in the boundary and resolution design for blocks in different conditions. The boundary length and resolution for each conditions are: $l_3 = 2m,\ r_3 = 0.02m,\ l_2 = 3m,\ r_2 = 0.2m,\ l_1 = 3m,\ r_1 = 0.2m$, and $l_0 = 8m,\ r_0 = 0.5m$. Clipping thresholds for different conditions are also designed for steady robust performance and less complexity consumption: $\alpha_3 = 2$, $\alpha_2 = 2$, $\alpha_1 = 10$ and $\alpha_0 = 10$ in log-domain.

8.2 Performance Analysis

8.2.1 Complexity Reduction

One of the major goal of this newly proposed algorithm is to reduce the huge complexity of original SPAWN, which mainly comes from fusing the messages at receiver agents. It can be divided into two aspects: number of total messages before performance converge and the size of the message.With the new cooperation scheme, the total number of messages has been largely reduced, this will be discussed in detail later in network traffic reduction and convergence speed. For the size of the message, clipping is the most powerful technique. After clipping, the size of new messages is only $10^{-3} - 6 \times 10^{-3}$ of the original one, as α goes from 1 to 10. Namely, as clipping is applied both on transmitter and receiver, the complexity of fusing one message is only $10^{-6} - 10^{-5}$ as the old one. Besides, geometrical pre-location and dynamic grid already reduce the original size of the beliefs, as messages, the complexity has been reduced completely.

We use $\mathcal{O}_{est}$ and $\mathcal{O}_{mp}$ presenting the complexity of initializing the belief $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$, and fusing message for each agent, accordingly. Let N_{an} and N_{ag} denote the average number of anchors and neighbor agents for one agent. n is the agent condition $n \in \{3,2,1,0\}$ R is the clipping rate, namely, the size of new message divide the size of original one. After applied to blocks, the clipping rate is around 3×10^{-2}. The formal analysis as follow:

- Original SPAWN

  $\mathcal{O}_{est} = \mathcal{O}(N_{an}(\frac{l}{r})^2)$, $\mathcal{O}_{mp} = \mathcal{O}(N_{ag}(\frac{l}{r})^4)$

- New algorithm

  $\mathcal{O}_{est} = \mathcal{O}(N_{an}^n(\frac{l_n}{r_n})^2)$,
  $\mathcal{O}_{mp} = \mathcal{O}((\frac{l_n}{r_n})^2 R_n(\sum_{i=n}^{3} N_{ag}^i(\frac{l_i}{r_i})^2 R_i))$

We can see the $\mathcal{O}_{mp}$ is the major item for both algorithms. With the parameters above, we can

roughly compare the complexity for one agent updating its belief one time. Total operation times for SPAWN is 3×10^{11} and for new algorithm is 3.2×10^{2}. With such small complexity, the time consumption for processing a network scenario is within 10 seconds for normal laptop yet original SPAWN takes hours or days.

8.2.2 Network Traffic Reduction and Convergence Speed

Besides complexity, heavy network traffic is another serious obstacle for SPAWN. Not only the large packets as messages, but also in every iteration, each agent broadcasts its belief. Former work has been proposed by using censoring to reduce the traffic [42, 43], our new algorithm achieve reduction by changing the cooperative scheme. As described in Section III, an agent will broadcast to all neighbors only when it becomes a reliable source. This procedure already contains ideas like transmitter/receiver censoring. In most scenarios, nearly all agents will only broadcast 1 or 2 times to achieve quite comparable performance with SPAWN. From the aspect of convergence speed, attractive performance is achieved by updating belief only 1 time for most agents, namely from $b_{\mathbf{x}_i}^{(0)}(\mathbf{x}_i)$ to $c_{\mathbf{x}_i}^{(1)}(\mathbf{x}_i)$. For the result of new algorithm to converge, it mostly only requires updating belief 1 time for all agents, while SPAWN needs at least 5 iterations to becoming converge in the scenarios used. Another advantage of the new algorithm is avoid setup iteration times manually, since it can autocratically give the satisfying solution for nearly all agents.

8.2.3 Positioning Performance

We collect the errors of all 40 scenarios for both original SPAWN and newly proposed algorithm and compared in Figure 3. Because of lacking further iterations, the performance is worse than SPAWN in the small error region. However, it can be improved by let the agents exchanging beliefs again after 1 new-defined iteration.

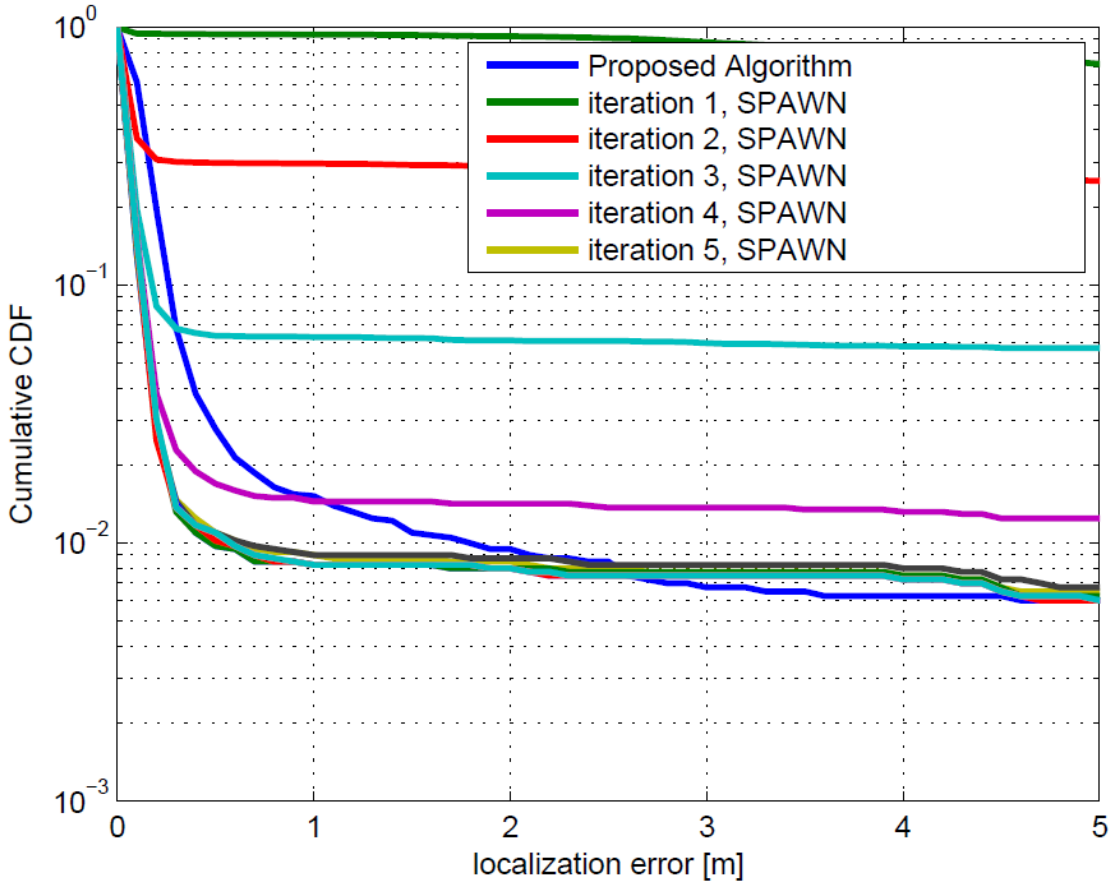

Figure 8.1: Performance comparison

The new algorithm offers large degrees of freedom considering the trade-off between complexity and accuracy of performance, which can be optimized by different requirements.

8.2.4 Sensitivity and Robustness to Parameters

The free design parameters of new algorithm are boundary length of block l_n, resolution of block r_n and clipping threshold α_n. Generally, the location error is not much affected if l_n is greater than 20% of its maximum communication range. The smaller l_n is, the more it depends on geometrical structure and estimation accuracy of the source agents. r_n have some influence in the error distribution in small error region. As for α_n, small α_n introduce more uncertainty which results in less reliable performance. However, before α_n drop below a

certain threshold, the performance stays nearly the same. The critical threshold of α_n is also related to l_n and α_n, which is decided by how many points, $\mathbf{x}_i$, left in φ_i after clipping.

Chapter 9

Conclusion and Future Work

Considering the shortcomings of SPAWN, we propose a new algorithm as a revolutionary solution. With three major techniques, geometrical pre-location, dynamic grid, and clipping, we successfully reduce the complexity completely and endue the capability to effectively process arbitrary large network scenarios. Together with the new cooperation scheme, the new algorithm also contribute to the network traffic reduction greatly. One attractive area of the future work is fully analysis and optimized parameters design for different network scenarios based on different accuracy and complexity requirements.

Bibliography

[1] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[2] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[3] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of pearl's "belief propagation" algorithm," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 140–152, 1998.

[4] S. Amari and H. Nagaoka, *Methods of information geometry*. Amer Mathematical Society, 2007, vol. 191.

[5] A. Grant, "Information geometry and iterative decoding," in *IEEE Communication Theory Workshop, Aptos, CA*, 1999.

[6] S. Stefan, "Information geometric analysis of iterative receiver structures," Master's thesis, Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, 2008.

[7] J. Walsh and P. Regalia, "Belief propagation, dykstra's algorithm, and iterated information projections," *Information Theory, IEEE Transactions on*, vol. 56, no. 8, pp. 4114–4128, 2010.

[8] R. Rockafellar, *Convex analysis*. Princeton Univ Pr, 1997, vol. 28.

[9] S. Amari, "Information geometry of the em and em algorithms for neural networks," *Neural networks*, vol. 8, no. 9, pp. 1379–1408, 1995.

[10] M. Moher and T. Gulliver, "Cross-entropy and iterative decoding," *Information Theory, IEEE Transactions on*, vol. 44, no. 7, pp. 3097–3104, 1998.

[11] F. Alberge, "Iterative decoding as dykstra's algorithm with alternate i-projection and reverse i-projection," in *EUSIPCO Proc*, 2008.

[12] S. Ikeda, T. Tanaka, and S. Amari, "Stochastic reasoning, free energy, and information geometry," *Neural Computation*, vol. 16, no. 9, pp. 1779–1810, 2004.

[13] I. Csiszár and P. Shields, *Information theory and statistics: A tutorial*. Now Publishers Inc, 2004.

[14] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[15] I. Csisz, G. Tusnády *et al.*, "Information geometry and alternating minimization procedures," *Statistics and decisions*, 1984.

[16] R. Dykstra, "An iterative procedure for obtaining *i*-projections onto the intersection of convex sets," *The Annals of Probability*, vol. 13, no. 3, pp. 975–984, 1985.

[17] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The annals of mathematical statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[18] L. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[19] H. Bauschke, P. Combettes, and D. Noll, "Joint minimization with alternating bregman proximity operators," *Pacific Journal of Optimization*, vol. 2, no. 3, pp. 401–424, 2006.

[20] H. Bauschke and J. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, vol. 79, no. 3, pp. 418–443, 1994.

[21] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 41–53, 2005.

[22] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks,[ieee signal process," 2005.

[23] D. Jourdan, D. Dardari, and M. Win, "Position error bound for uwb localization in dense cluttered environments," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 44, no. 2, pp. 613–628, 2008.

[24] D. Davide, L. Jaime, Z. Moe *et al.*, "The effect of cooperation on localization systems using uwb experimental data," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.

[25] H. Wymeersch, U. Ferner, and M. Win, "Cooperative bayesian self-tracking for wireless networks," *Communications Letters, IEEE*, vol. 12, no. 7, pp. 505–507, 2008.

[26] G. Mao and B. Fidan, *Localization algorithms and strategies for wireless sensor networks*. Information Science Reference-Imprint of: IGI Publishing, 2009.

[27] A. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 24–40, 2005.

[28] S. Gezici, "A survey on wireless position estimation," *Wireless Personal Communications*, vol. 44, no. 3, pp. 263–282, 2008.

[29] X. Li, "Rss-based location estimation with unknown pathloss model," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 12, pp. 3626–3633, 2006.

[30] A. D'Amico, U. Mengali, and L. Taponecco, "Energy-based toa estimation," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 3, pp. 838–847, 2008.

[31] S. Venkatraman and J. Caffery Jr, "Hybrid toa/aoa techniques for mobile location in non-line-of-sight environments," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 1. IEEE, 2004, pp. 274–278.

[32] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–426, 2009.

[33] S. Kay, "Fundamentals of statistical signal processing: estimation theory," 1993.

[34] X. Sheng and Y. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 53, no. 1, pp. 44–53, 2005.

[35] G. Destino and G. Abreu, "Reformulating the least-square source localization problem with contracted distances," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*. IEEE, 2009, pp. 307–311.

[36] K. Ho, X. Lu, and L. Kovavisaruch, "Source localization using tdoa and fdoa measurements in the presence of receiver location errors: Analysis and solution," *Signal Processing, IEEE Transactions on*, vol. 55, no. 2, pp. 684–696, 2007.

[37] X. Sheng and Y. Hu, "Energy based acoustic source localization," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 551–551.

[38] S. Gezici, I. Guvenc, and Z. Sahinoglu, "On the performance of linear least-squares estimation in wireless positioning systems," in *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE, pp. 4203–4208.

[39] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[40] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and vlsi implementation," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 2, pp. 290–300, 2008.

[41] S. Schwandter, P. Fertl, C. Novak, and G. Matz, "Log-likelihood ratio clipping in mimo-bicm systems: Information geometric analysis and impact on system capacity," 2009.

[42] K. Das and H. Wymeersch, "Censored cooperative positioning for dense wireless networks," in *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on*. IEEE, 2010, pp. 262–266.

[43] ——, "A network traffic reduction method for cooperative positioning," in *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on*. IEEE, 2011, pp. 56–60.