國立臺灣大學電機資訊學院資訊工程研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

中文標題，請到ntuvars.tex輸入你的資料

MsWave

王瑞斌

Jui-Pin Wang

指導教授：林守德博士

Advisor: Shou-De Lin, Ph.D.

中華民國 103 年 7 月

July, 2014

國立臺灣大學
資訊工程研究所

碩士論文

中文標題，請到ntuvars.tex輸入你的資料

王瑞斌

撰

103
7

國立臺灣大學（碩）博士學位論文
口試委員會審定書

論文中文題目
論文英文題目

本論文係○○○君（○學號○）在國立臺灣大學○○學系、所完成之碩（博）士學位論文，於民國○○年○○月○○日承下列考試委員審查通過及口試及格，特此證明

口試委員：

______________________________（簽名）

（指導教授）

________________　________________

________________　________________

________________　________________

________________　________________

系主任、所長 ________________________（簽名）

（是否須簽章依各院系所規定）

致謝

這裡將簡單介紹如何利用 LaTeX 來編輯你的畢業論文，若不知道 LaTeX 是什麼或是沒有概念的話，建議你可以簡單看過放在此資料夾裡的李果正 -大家來學 LaTeX前四章內容，在下載適合的 LaTeX 整合發行套件之後（請看第 III.項），可以嘗試用剛安裝好的 LaTeX 編輯器來編譯thesis.tex這份文件，編譯的方法可以看下面第 V.項的介紹，若編譯成功，所編譯出來的 thesis.pdf 文件的應該會跟此 demo.pdf 文件一模一樣，而且沒有任何問號符號，走到這一步的話，就差不多可以開始邊學習 LaTeX 邊編輯你的畢業論文了！基本上會使用到的指令都包含在論文的的各章節裡，怎麼在論文裡寫公式或是放圖之類的就自行看 tex 檔學吧。如果有任何問題或建議可以來信與我討論，我的信箱是dran31545@gmail.com，或是到此範本Google Project裡面的Issues貼上你的問題與建議，我會盡我所能更新此範本，也歡迎大家自行重製、改良此範本並散布給他人，祝大家順利畢業！

要編輯致謝請打開acknowledgementsCH.tex

I. 此範本參考並修改自下列網站的資料：

- 如何用 LaTeX 排版臺灣大學碩士論文
  —台灣大學論文 LaTeX 樣版原創者黃子桓的教學網頁
- LaTeX 常用語法及論文範本
  —Hitripod所修改的範本，這裡參考了許多他所寫的格式和內容
- 使用 LaTeX 做出精美的論文
- XeTeX：解決 LaTeX 惱人的中文字型問題
- 台灣大學碩士、博士論文的 Latex 模板

II. 幾個有用的參考資料及網路資源：

- 李果正 -大家來學 LaTeX—建議先看完前四章
- WIKIBOOKS-LaTeX—好用的線上工具書
- Working with a .bib file using JabRef
- Using BibDesk - A short tutorial
- LaTeX for Physicists

III. 下載 LATEX 整合發行套件，可參考TeX Collection：

1. MacTeX: For **MacOSX**，下載MacTeX.pkg
2. ProTeXt: For **Windows**，下載ISO file
3. TeX Live: For **GNU/Linux** and **MacOSX**, and **Windows**，下載ISO file
4. CTAN: The Comprehensive TeX Archive Network.

IV. 好用的程式：

- 文獻管理系統：
  1. JabRef
     可參考Working with a .bib file using JabRef或是Google及YouTube
  2. BibDesk (For Mac)
     可參考Using BibDesk - A short tutorial或是Google及YouTube
- 方程式編輯器：Daum Equation Editor (Chrome App，必須使用 Google 瀏覽器)

V. 編譯流程：

1. `xelatex thesis`
   對 thesis.tex 進行第一次 XeLaTeX 編譯，產生 thesis.pdf 以其他檔案
2. `bibtex thesis`
   對 thesis.tex 進行 BibTeX 編譯，產生 bbl 檔以及 blg 檔
3. `xelatex thesis`
   對 thesis.tex 進行第二次 XeLaTeX 編譯，產生目錄、圖表連結及參考文獻
4. `xelatex thesis`
   對 thesis.tex 進行第三次 XeLaTeX 編譯，產生參考文獻連結，完成編譯

注意！此範本使用 cite 套件，可依據你利用文獻管理系統所整理好的thesisbib.bib檔在論文最後產生參考文獻頁面，若你的系所規定要在每個章節的後面產生參考文獻，則可以用 chapterbib 套件，來對每個有附參考文獻的章節 tex 檔進行一次 BibTeX 編譯產生 bbl 檔，如範例的introduction.tex、THM.tex和EXP.tex，如果有這需要請把thesis.tex檔裡使用 cite 套件的指令利用註解符號% 來取消使用 cite 套件，並刪去出現在使用 chapterbib 套件指令前面的註解符號% 來啟動使用 chapterbib 套件

```
\usepackage{cite}
%\usepackage{chapterbib}
```

改成

```
%\usepackage{cite}
\usepackage{chapterbib}
```

再來利用註解符號% 取消會把參考文獻放在論文最後的指令

```
\bibliographystyle{unsrt}
\addcontentsline{toc}{chapter}{\bibname}
\bibliography{thesisbib}
```
改成
```
%\bibliographystyle{unsrt}
%\addcontentsline{toc}{chapter}{\bibname}
%\bibliography{thesisbib}
```

再把用來輸入章節檔案的 \input 指令改成 \include 指令

```
\input{introduction}  =>  \include{introduction}
\input{THM}           =>  \include{THM}
\input{EXP}           =>  \include{EXP}
```

最後記得在每個有附參考文獻的章節加上產生參考文獻的指令，即在introduction.tex、THM.tex和EXP.tex三個檔案裡最後啟動下面兩行指令

```
%\bibliographystyle{unsrt} => \bibliographystyle{unsrt}
%\bibliography{thesisbib}  =>  \bibliography{thesisbib}
```

而編譯時則需要對有附參考文獻的introduction.tex、THM.tex和EXP.tex各做一次 BibTeX 編譯，編譯流程如下

1. `xelatex thesis`
   對 thesis.tex 進行第一次 XeLaTeX 編譯，產生 thesis.pdf 及其他檔案
2. `bibtex introduction`
   對 introduction.tex 進行 BibTeX 編譯，產生 bbl 檔以及 blg 檔
3. `bibtex THM`
   對 THM.tex 進行 BibTeX 編譯，產生 bbl 檔以及 blg 檔
4. `bibtex EXP`
   對 EXP.tex 進行 BibTeX 編譯，產生 bbl 檔以及 blg 檔
5. `xelatex thesis`
   對 thesis.tex 進行第二次 XeLaTeX 編譯，產生目錄、圖表連結及參考文獻

6. `xelatex thesis`
   對 thesis.tex 進行第三次 XeLaTeX 編譯，產生參考文獻連結，完成編譯

VI. 補充說明與注意事項：

- 口試委員會審定書：
  請到台大圖書館網頁的電子論文服務下載論文格式範本，並修改成正確的格式，也可到此範本所在資料夾的cert.doc修改。當然你也可以利用 LaTeX 來編輯，你只要填好ntuvars.tex檔的資料，並去除在 thesis.tex 裡下面這行的註解符號%

  ```
  %\makecertification
  ```

  編譯完後就可以產生審定書格式。口試通過後，請把已經簽名的審定書掃描成 pdf 檔，再取代原本的cert.pdf，即可放上已簽名的審定書。處理審定書出現的指令在 thesis.tex 裡

  ```
  %----------- generate the certification ...
  %\makecertification
  %----------- includepdf by using package ...
  \addcontentsline{toc}{chapter}{口試委員會審定書}
  \includepdf[pages={1}]{cert.pdf}
  ```

- 浮水印：
  資料夾已經附上浮水印檔案了，若學校有更改，到請到台大圖書館網頁的電子論文服務下載pdf 格式的浮水印到此範本所在資料夾。若要開啟關閉浮水印功能，即自行刪去或加上下面位於thesis.tex指令的註解符號%

  ```
  %\CenterWallPaper{0.174}{watermark.pdf}
  %\setlength{\wpXoffset}{6.1725cm}
  %\setlength{\wpYoffset}{10.5225cm}
  ```

- 單面印刷與雙面印刷：
  此範本為單面印刷，若論文頁數超過 80 頁，依規定需要用雙面印刷，此時只需把 thesis.tex 裡的

  ```
  \documentclass[a4paper, 12pt, oneside]{book}
  ```
  改成
  ```
  \documentclass[a4paper, 12pt, twoside]{book}
  ```

- 如何加入附錄?

在thesis.tex裡，依需求選擇 input 或 include，刪去% 符號來輸入附錄章節

```
%----------- Input your appendix here  -----------
%\input{AppendixA}
%or %chapter cite  == \include
%\include{AppendixA}
```

在章節檔 AppendixA.tex 裡，開頭打

```
\chapter{First appendix title}
```

即可，以此類推。

- 系上規定論文圖表須全部放到最後獨立出來的章節，且章節不出現在目錄中：

在thesis.tex裡，依需求選擇 input 或 include，刪去% 符號來輸入圖表章節

```
%----------- Input your Figure chapter here  -----------
%\input{EndFigTab}
%chapter cite  == \include
%\include{EndFigTab}
```

在章節檔EndFigTab.tex裡有範例和說明可供參考，要注意正文的圖表和附錄的圖表要分清楚，即在EndFigTab.tex內

```
\renewcommand{\thefigure}{\arabic{chapter}.
\arabic{figure}}
\renewcommand{\thetable}{\arabic{chapter}.
\arabic{table}}
%--- Input your main figures and tables here  ---
```

這幾行之後章節計數器格式已切換為 1...9，放正文的圖表，

```
\renewcommand{\thefigure}{\Alph{chapter}.
\arabic{figure}}
\renewcommand{\thetable}{\Alph{chapter}.
\arabic{table}}
%--- Input your appendix figures and tables here  ---
```

這幾行之後章節計數器格式已切換為A...Z，放附錄的圖表。另外要取消圖表的浮動功能，才能讓圖表按照指令出現順序排好，即把平常使用的圖表指令

```
\begin{figure}[htb]
...
\begin{table}[htb]
```

改成

```
\begin{figure}[!]
...
\begin{table}[!]
```

剩下的只要注意章節圖表的計數器設定即可。\ref 和 \label 指令可以在此圖表章節與正文章節使用。

- 如果我想要修改 margin(文字邊界) 的話，可以從哪裡下手呢?
  請打開ntu.sty修改下面這行的上下左右參數即可：

  ```
  \RequirePackage[top=3cm,left=3cm,bottom=2cm,right=3cm]
  {geometry}
  ```

- 我想引用 Twomey (1974): Pollution and planetary albedo 這篇論文，如何用 \cite 引用它的時候在內文顯示 Twomey (1974) [編號] ？
  建議使用 natbib 套件，參考資料如下：
  LaTeX/Bibliography Management
  Overview of Bibtex-Styles
  Reference sheet for natbib usage
- XƎTEX ：
  此範本中文字體使用XƎTEX 轉換，細節請參考Hitripod寫的XeTeX：解決 LaTeX 惱人的中文字型問題。
- 如何輸入英文 ‘單引號’ 和 “雙引號” 以及不同長度的破折號？
  可以參考李果正 -大家來學 LATEX第 17 頁針對標點符號的遊戲規則，範例如下，輸入以下指令：

  ```
  `單引號'\\
  ``雙引號''\\
  -hyphen\\
  --en-dash\\
  ---em-dash\\
  ```

  則顯示：
  ‘單引號’
  “雙引號”

-hyphen

–en-dash

—em-dash

中文摘要

請打開並編輯abstractCH.tex
關鍵字：壹、貳、參、肆、伍、陸、柒

Abstract

Open and edit abstractEN.tex

Key words:A, B, C, D, E, F, G

Contents

List of Figures

List of Tables

Chapter 1

Get started with LaTeX

Three common font styles in this text:

- **Item1**: *Italic 中文 123*
- **Item2**: **Bold 中文 123**
- **Item3**: *slant 中文 123*

About the advance latex grammer see the next section 1.1.

1.1 LaTeX Adavanced Features

The following features would be introduced in the coming subsections:

- SubSection 1.1.1: **Figure**
- SubSection 1.1.3: **Verb**
- SubSection 1.1.3: **Verb**
- SubSection 1.1.4: **Enumeration**
- SubSection 1.1.2: **Table**
- SubSection 1.1.5: **Code Display**
- SubSection 1.1.6: **Math**

- SubSection 1.1.7: **Algorithms**

1.1.1 Figure

Figure 1.1: A picture of a tiger.

Figure 1.1 is a picture of a tiger.

1.1.2 Table

Table examples on WIKIBOOKS.

Table 1.1: Table Example 1

Start	End	Character Block Name
3400	4DB5	CJK Unified Ideographs Extension A
4E00	9FFF	CJK Unified Ideographs

Table 1.2: Table Example 2

Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Table 1.3: Table Example 3

Allocation	Allocation, Element, Type, Script
Data Types	Byte2, Byte3, and Byte4 Float2, Float3, Float4 Int2, Int3, Int4 Long2, Long3, Long4 Matrix2f, Matrix3f, Matrix4f Short2, Short3, Short4
Graphics	Mesh ProgramFragment, ProgramRaster ProgramStore, ProgramVertex RSSurfaceView

Table 1.4: Table Example 4

Team sheet		
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucus Radebe
	DC	Michael Duberry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Table 1.5: Table Example 5

Team	P	W	D	L	F	A	Pts
Manchester United	6	4	0	2	10	5	12
Celtic	6	3	0	3	8	9	9
Benfica	6	2	1	3	7	8	7
FC Copenhagen	6	2	1	2	5	8	7

1.1.3 Verb

Let’s take a overview on how to type special characters:

<FRAMEWORKS_BASE>/graphics/java/android/renderscript

[1] You could also go back to the beginning of the chapter by the **hyperref**.

1.1.4 Enumeration

1. Enumerated Item1
2. Enumerated Item2
3. Enumerated Item3

1.1.5 Code Display

Here is a ”Hello, DanDing.” example:

```
void main(int argc, char **argv)
{
    printf(” ´  _> `  ”);
}
```

Another example with line numbers:

```
void main(int argc, char **argv)
{
    printf(” ´  _> `  ”);
}
```

Matlab example:

```
function y = demo(x) % This is a comment.
    str = 'hello there';
    y = x + 1;
end
```

[1] Path of <APP_intermediates>: <ANDROID_ROOT>/ out/ target/ common/ obj/ APPS/ APP-NAME_intermediates/

1.1.6 Math

- Inline mode:

  The solution to $\sqrt{x} = 5$ is $x = 25$.

- Display mode:

  The solution to

  $$\sqrt{x} = 5$$

  is

  $$x = 25.$$

- Numbered mode:

  $$2 + 2 = 4 \tag{1.1}$$

- Non-numbered:

  $$2 + 2 = 4$$

- Aligning:

  $$\begin{aligned} 2x^2 + 3(x-1)(x-2) &= 2x^2 + 3(x^2 - 3x + 2) \\ &= 2x^2 + 3x^2 - 9x + 6 \\ &= 5x^2 - 9x + 6 \end{aligned}$$

- Fractions:

  $$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

- Matrix:

  $$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

More examples on WIKIBOOKS.

1.1.7 Algorithms

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \lor x \neq 0$
Ensure: $y = x^n$

```
y ⇐ 1
if n < 0 then
    X ⇐ 1/x
    N ⇐ −n
else
    X ⇐ x
    N ⇐ n
end if
while N ≠ 0 do
    if N is even then
        X ⇐ X × X
        N ⇐ N/2
    else[N is odd]
        y ⇐ y × X
        N ⇐ N − 1
    end if
end while
```

More examples on WIKIBOOKS.

Chapter 2

Introduction

HiHi Iam r44 . The organization of this thesis is as follows. In chapte **??**, the theoretical background and definition of surface plasmon will be included [**?**]. Chapte 5 contains description of experiment methods such as atomic force microscopy and scanning electron microscopy.

2.1 Thesis Overview

In this section, we describe the overview of this thesis.

Chapter 3

Related Works

HiHi Iam r44 . The organization of this thesis is as follows. In chapte **??**, the theoretical background and definition of surface plasmon will be included [**?**]. Chapte 5 contains description of experiment methods such as atomic force microscopy and scanning electron microscopy.

‘

Chapter 4

Methodology

HiHi Iam r44 . The organization of this thesis is as follows. In chapte **??**, the theoretical background and definition of surface plasmon will be included [1]. Chapte 5 contains description of experiment methods such as atomic force microscopy and scanning electron microscopy.

4.1 Problem Setup

There are a query set $Q = \{q_1, q_2, ..., q_T\} \subset \mathbb{R}^D$ at the server P and a dataset $X_i \subset \mathbb{R}^D$ on each local machine M_i. For each coming query q_t, we want to find its k_{th} nearest neighborhood among these distributed datasets while reducing the transmission cost between P and each M_i.

4.2 Overview of Our Framework

In this section, we describe the overall framework of our work. Then, we will give the details about the framework in the following sections.

There are two main phases in our framework. For each X_i, the first phase only needs to be done for once. On the other hand, we need to run the second phase for each new query q_t.

The first phase is an preprocessing procedure for the second phase. Its goal is to improve the performance of pruning in the second phase. We will prove in the section 4.5.2 that this pruning power is highly correlated to the distribution of the norm of the feature vectors. As a result, each M_i would learn an othorogal matrix W_i for its X_i to fit our desired distribution and then send each W_i back to P. We can notice that this phase is only dependent on X_i and independent of q_t. Therefore, we only need to do the first phase for once. we give the details about how to learn W_i, how to send it back to P in the section 4.3.

The second phase is the main procedure of our framework. Note that P have already got W_i for each X_i in the beginning of the second phase. For each coming query q_t, we iteratively prune some candidates which are impossible to be the kNN of q_t to reduce the search space until there are only k candidates left.

To prune candidates iteratively, we divide the second phase into several rounds. For each round j, we use a $Select$ function $S_i(q_t, j; \theta_t)$ to generate the values for trasmitting from P to M_i, where S_i is the importance-selecting function of M_i and θ_t is its parameters for q_t. (We put the details of S_i at the section **??**.) By these values, each M_i could calculate the bounds between each candidate x_l and q_t. With these bounds, P would be able to determine which candidates are definitely not our answer and then disregards them in the following rounds. By these pruning, we could achieve the goal of saving transmission cost from avoiding to consider the unnecessary candidates.

Note that we could use the square of the Euclidean distance instead of the origin Euclidean distance to find kNN as it is non-negative. So we will use the former one in our framework.

4.3 Orthogonal Transformation

Since the In this section, we describe the overview of this thesis.

4.3.1 Definition of Ortohogonal Transformation

Definition 1. *A matrix $W \in \mathbb{R}^{D\times D}$ is orthogonal if whose columns and rows are orthogonal vectors, i.e.*

$$W^T W = W W^T = I$$

where I is the identity matrix.

4.3.2 Property of Orthogonal Transformation

Property 1. *Let $x, y \in \mathbb{R}^{D\times 1}$, and $W \in \mathbb{R}^{D\times D}$ be an orthogonal matrix. Then,*

$$Dist(x,y) = \sum_{d=1}^{D}(x[d]-y[d])^2 = \sum_{d=1}^{D}(W[d,:]x - W[d,:]y)^2 = Dist(Wx, Wy)$$

where $W[d,:]$ is the d_{th} row of W.

We will use this important property in the section **??**.

4.4 Enhance the Bounds by the Orthogonal Transformation

In the section 4.2, we mentioned that the first phase is an auxiliary step for the second phase. After the introduction of the orthogonal transformation, we introduce this powerful tool into the first phase in our framework.

4.4.1 The Goal of the First Phase

Our goal in the first phase is to reduce the ranges of the bounds used in the second phase. Since we will use a threshold to prune the impossible candidates according to their bounds in the The pruning procedure, the ranges of the bounds would be one of the most influential factor of the pruning power.

Suppose that we want to prune some candidates whose lower bounds are upper than a threshold thr. In these figures, we could see that when the ranges of their bounds is short, more candidates would be pruned than those with long ranges of the bounds. In other words, the shorter the range of the bound, the higher chance this candidate would be pruned if it is not our final answer of kNN.

4.4.2 Definition of the Bounds

First, we need to define the bounds for the pruning. Recall that given a query q_t, our goal is to find its kNN in these distributed datasets X_i. Intuitively, we need to calculate the square of the Euclidean distance $Dist(q_t, x), \forall x \in \cup_i X_i$. However, to cacluate $Dist(q_t, x)$, we need to send the whole q_t to the local machines or send the whole x to P, which causes a huge transmission cost. Therefore, instead of the exact value of Euclidean distances, our propsed framework uses bounds to find the kNN .

Definition 1. *$\forall x, y \in \mathbb{R}^D$, a lower bound $LB(x, y)$ and a upper bound $UB(x, y)$ must satisfy the following inequation:*

$$LB(x, y) \leq Dist(x, y) \leq UB(x, y)$$

4.4.3 Relation Between the Norms and the Bounds

To acheive the goal of reducing the length of the ranges of the bounds, we could look into the derivation of the bounds. Suppose there are two vectors $x, y \in \mathbb{R}^D$, but we could only observe the first s dimensions of x and $\sum_{d=s+1}^{D} x[d]^2$, which is the square of two norm of the unobserved part $x[s+1 : D]$. In the section 4.5.2, we give the bounds as

$$\begin{aligned} LB(x, y) &= \sum_{d=1}^{s} (x[d] - y[d])^2. \\ UB(x, y) &= \sum_{d=1}^{s} (x[d] - y[d])^2 \\ &+ \sum_{d=s+1}^{D} x[d]^2 + \sum_{d=s+1}^{D} y[d]^2 \\ &+ 2 \times \sqrt{\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2}. \end{aligned}$$

Therefore, we could get the length of the range by the substraction.

$$Len = UB(x,y) - LB(x,y) = \sum_{d=s+1}^{D} x[d]^2 + \sum_{d=s+1}^{D} y[d]^2 + 2 \times \sqrt{\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2}.$$

Since the term $\sum_{d=s+1}^{D} x[d]^2$ is given, all we could do is to reduce the length with the help of the $\sum_{d=s+1}^{D} y[d]^2$ term. If we could reduce $\sum_{d=s+1}^{D} y[d]^2$, the term $\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2$ would also decrease and then make Len smaller. Therefore, our goal now becomes to make the term $\sum_{d=s+1}^{D} y[d]^2$ as small as possible, which is the square norm of the vector $y[s+1:D]$.

Note that the lower (upper) bounds is non-decreasing (non-increasing) as s becomes larger and $LB(x,y) = UB(x,y)$ when $s = D$. This means that LB and UB would be exactly equal to $\sum_{d=1}^{D} (x[d] - y[d])^2$ eventually.

To reduce the length of the ranges Len for each s, we hope to make $\sum_{d=s+1}^{D} y[d]^2$ as small as possible. However, since the feature vector y is given from datasets, the value of $\sum_{d=s+1}^{D} y[d]^2$ is already determined when s is given. As a result, we introduce the orthogonal transformation to achieve this goal.

4.4.4 Equivalent Bounds After Transformation

From the section 4.3.2, we know the distance of two vectors won't be changed after an orthogonal transformation. Now we use this property to achieve our goal to reduce the length of the ranges Len given s.

Given an orthogonal transformation $W \in \mathbb{R}^{D \times D}$, we have $Dist(x,y) = Dist(Wx, Wy)$. This means that the bounds we derivated before could also be the bounds for $Dist(Wx, Wy)$. That is,

$$LB(x,y) \leq Dist(x,y) = Dist(\hat{x}, \hat{y}) \leq UB(x,y)$$

where $\hat{x} = Wx$.

This also means that we could use the same way to derivate the lower bounds and upper bounds for $Dist(Wx, Wy)$ and these bounds are also the bounds for $Dist(x,y)$.

That is,

$$LB(\hat{x}, \hat{y}) \leq Dist(x, y) = Dist(\hat{x}, \hat{y}) \leq UB(\hat{x}, \hat{y})$$

So, we could use the bounds $LB(\hat{x}, \hat{y}), UB(\hat{x}, \hat{y})$ for $Dist(x, y)$ and the length of range we want to reduce becomes

$$\hat{Len}(W) = UB(\hat{x}, \hat{y}) - LB(\hat{x}, \hat{y}) = \sum_{d=s+1}^{D} \hat{x}[d]^2 + \sum_{d=s+1}^{D} \hat{y}[d]^2 + 2 \times \sqrt{\sum_{d=s+1}^{D} \hat{x}[d]^2 \times \sum_{d=s+1}^{D} \hat{y}[d]^2}.$$

which becomes a function of W.

As a result, instead of trying to reduce $\sum_{d=s+1}^{D} y[d]^2$ which is impossible as we mentioned in the section 4.4.3, our goal becomes to reduce $\sum_{d=s+1}^{D} \hat{y}[d]^2$ with the help of W.

4.4.5 Reduce the Norm with Orthogonal Transformation

For $y \in \mathbb{R}^{D \times 1}$ and $W \in \mathbb{R}^{D \times D}$, given s, we want to reduce $\sum_{d=s+1}^{D} \hat{y}[d]^2$ as much as possible. However, since s is unknown while deciding W in the first phase of our framework, we have to handle all possible values which s could be. Moreover, since $\sum_{d=1}^{D} \hat{y}[d]^2$ is equal to $\sum_{d=1}^{D} y[d]^2$, which is independent with W, if the $\sum_{d=s+1}^{D} \hat{y}[d]^2$ decreases with some W, the term $\sum_{d=1}^{s} \hat{y}[d]^2$ must increase. Here, we use a more general strategy to deal with these problems.

We could look the term $\sum_{d=s+1}^{D} \hat{y}[d]^2$ from a different angle. Actually, this term is the square norm of the latter part of the vector $\hat{y}$. Therefore, although $\sum_{d=1}^{D} \hat{y}[d]^2$ is a constant for W, we could reduce the square norm of the latter part by increasing the forward part of it. In other words, we move the norm of the latter part of y to its forward part. To accomplish it, we design an objective function and then optimize this function to find our ideal W.

$$f(W; y) = \sum_{d=1}^{D} w_d \times \hat{y}[d]^2 = \sum_{d=1}^{D} w_d \times (W[d,:]y[d])^2 \tag{4.1}$$

where $w_d = d, \forall d = 1 : D$.

Because w_d would give the larger penalty as d increases, the elements in the latter part of $\hat{y}$ would be forced to become small while minimizing this objective function. This is exactly our goal to reduce $\sum_{d=s+1}^{D} \hat{y}[d]^2$. Therefore, our question becomes how to optimize this objective function with the constraints that W must be an orthogonal matrix.

4.4.6 Optimize with Orthogonal Constraints

Finally, we could introduce this concept of reducing the norms into our framework. In the first phase, we solve the following optimization problem to learn an orthogonal matrix W_i for each machine Mi.

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & F_i(W) \\ \text{subject to} \quad & W^T W = W W^T = I \end{aligned} \tag{4.2}$$

where

$$F_i(W) = \sum_{x \in X_i} f(W; x) = \sum_{x \in X_i} \sum_{d=1}^{D} d \times (W[d,:]x[d])^2 \tag{4.3}$$

.

This is an optimization problem with constraints that its solution must be an orthogonal matrix. We could solve it efficiently with the help of the package of [2] as long as we have its gradient.

Put the gradient, one line version.

After we get the optimal W_i^* for each M_i, we send these matrices back to the server P. Since the learning of W_i^* is independent with the queries in the future, we only have to go through the procedure of learning W_i^* for once if X_i doesn't change too much.

4.4.7 Reduce the Cost of Sending Matrices

http://math.stackexchange.com/questions/375344/parameters-to-represent-degrees-of-freedom-in-n-times-n-orthogonal-real-matric

http://math.stackexchange.com/questions/28189/freedoms-of-real-orthogonal-matrices

4.5 Prune by the Bounds

Now we start to discuss the second phase of our framework. In this section, we talk about how to prune the candidates if we already have bounds. Note that this mechanism is the most crucial part to achieve our goal to save the transmission cost.

4.5.1 Prune the Candidates with the Bounds

For the query q_t, if we already know $LB(q_t, x)$ and $UB(q_t, x)$ $\forall x \in \cup_i X_i$, we could use the k_{th} smallest upper bounds and directly prune those x whose lower bounds are higher than this value thr. I.e., we want to prune

$$\{x | LB(q_t, x) > thr, \forall x \in \cup_i X_i\}$$

where thr is the k_{th} largest $UB(q_t, x)$ $\forall x \in \cup_i X_i$.

Here is an example of the pruning procedure.

TODO: Draw the figure about pruning.

We will talk about the details to generate these bounds and find the threshold in the section 4.5.4.

4.5.2 Derivation of the Bounds

Suppose there are two vectors $x, y \in \mathbb{R}^D$, we know the square of their Euclidean distance is

$$Dist(x, y) = \sum_{d=1}^{D} (x[d] - y[d])^2 \tag{4.4}$$

However, if we could only observe the first s dimensions of x, we could decompose their distance as

$$Dist(x, y) = \sum_{d=1}^{D} (x[d] - y[d])^2 = \sum_{d=1}^{s} (x[d] - y[d])^2 + \sum_{d=s+1}^{D} (x[d] - y[d])^2. \tag{4.5}$$

Since the first component of (4.5) is already known, all we need to do is to deal with the second term. Therefore, we further expand the second term as below:

$$\sum_{d=s+1}^{D}(x[d]-y[d])^2 = \sum_{d=s+1}^{D} x[d]^2 + \sum_{d=s+1}^{D} y[d]^2 - \sum_{d=s+1}^{D} 2\times x[d]\times y[d].$$

By this analysis, we find the final term is the inner product between two partial vector $x[s+1:D]$ and $y[s+1:D]$, which could be approximated by Cauchy–Schwarz inequality

$$\sum_{d=s+1}^{D} x[d]\times y[d] \leq \sqrt{\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2}. \tag{4.6}$$

After combing with (4.5) and (4.6), we derive the bounds as

$$LB(x,y) = \sum_{d=1}^{s}(x[d]-y[d])^2. \tag{4.7}$$

$$\begin{aligned} UB(x,y) &= \sum_{d=1}^{s}(x[d]-y[d])^2 \\ &+ \sum_{d=s+1}^{D} x[d]^2 + \sum_{d=s+1}^{D} y[d]^2 \\ &+ 2\times\sqrt{\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2}. \end{aligned} \tag{4.8}$$

We could notice that the calculation of the bounds only needs the first s dimensions of x and $\sum_{d=s+1}^{D} x[d]^2$. Therefore, we only need one more number to get the bounds for the unobserved part $x[s+1:D]$.

4.5.3 Calculation of the Bounds

After the derivation of the bounds, we describe the procedure of cacluating them in our framework.

For the query q_t, at the first round (i.e. $j=1$), P sends the first s_1 dimensions of q_t and $\sum_{d=s_1+1}^{D} q_t[d]^2$ to each M_i . With these values, each M_i would able to calculate the lower bounds $LB(q_t,x)$ and upper bounds $UB(q_t,x)$ for each $x\in X_i$. Then, after P getting the k_{th} smallest upper bounds as thr, we could run the pruning procedure.

In each following round (i.e. $j>1$), P sends the next s_j dimensions of q_t to each M_i whose instances were not pruned completely. These M_i will update their bounds as follows:

$$LB_j(x,y) = LB_{j-1}(x,y) + \sum_{d=p_{j-1}+1}^{p_j} (x[d]-y[d])^2. \tag{4.9}$$

$$\begin{aligned} UB_j(x,y) &= LB_j(x,y) \\ &+ \sum_{d=p_j+1}^{D} x[d]^2 + \sum_{d=p_j+1}^{D} y[d]^2 \\ &+ 2 \times \sqrt{\sum_{d=s+1}^{D} x[d]^2 \times \sum_{d=s+1}^{D} y[d]^2}. \end{aligned} \tag{4.10}$$

where $p_j = \sum_{i=1}^{j} s_i$, LB_j and UB_j indicate the lower bounds and upper bounds at the round j repectively.

We call those p_j as pivots, which mean that each machine would observe the first p_i elements of q_t at the round j.

4.5.4 Find the Threshold in Distributed Machines

The question now is to find the threshold thr for pruning. In [3], we directly send these bounds computed in each M_i back to the server P. However, it would make the transmission cost grow linearly with the number of total instances in these distributed machines and lead to expensive cost when our dataset is extremely huge. As a result, we propose a method that could make the growth of the cost independent with the number of total instances.

To be simplified, we could think this problem as follows: given many distribued numbers N_i, we want to find the k_{th} largest number among these N_i. The N_i here actually means the upper bounds at the machine M_i in our framework. Once we model this problem as this, we could solve it through modifying the work of [4].

In [4], there are also many phases to find the kNN. In the first phase, the server P would send the whole query to every machine. Then, in the following phases, it just focuses on finding the instances with the k_{th} largest distance with the query. To make it fit our problem, we can only use the phases of PRP except the first phase to find the k_{th}

largest upper bound as our threshold thr. Its cost is linear to

$$|M| \times (\left\lfloor \frac{k}{|M|} \right\rfloor + 1) = |M| + k,$$

which is much lower than [3] when the number of total instances is very large.

4.6 Decide the Pivots

The remaining problem is to decide how many dimensions (i.e. s_j) of q_t we have to send from the server P in each round j. If we send too few dimensions, the bounds would be too loose to prune any candidates and we will spend much unnecessary cost in finding the thresholds. On the other hand, if we send too many dimensions, although it could allow us to prune many candidates at once, it would send too many dimensions to some candidates which could be pruned by much fewer dimensions. That would also lead to the waste the transmission cost. Therefore, in this section, we propose a simple but effective method to decide how many dimensions of q_t we should send from P in each round.

4.6.1 Estimate the Number of Residual Machines

From the discussion above, we could notice that the decision of pivots is highly dependent on the transmission cost. Therefore, if we could estimate the cost as a cost function of the pivots p_j, we could decide these pivots by optimizing this function.

However, to estimate the cost, we need to know the number of residual candidates sites before sending those dimensions of q_t in each round. But it is almost impossible to know how many sites would be left after we send part of q_t before we actually send the part of q_t. Therefore, we estimate a vector called $EstResMach \in \mathbb{R}^D$ where $EstResMach[j]$ indicates our estimate of the number of residual machines *after* we send $q_t[1:j]$ to each local machine.

To allow P to estimate this vector $EstResMach$ without causing any more transmission cost, we use the history information from $q_1, ..., q_t$. Suppose that we just finish finding the answer for q_t, during the procedure, we would collect some such pairs of information

$(index_j, ResMach_j)$ for some j where $index_j$ means each candidate machine would observe the first $index_j$ dimensions of q_t at the round j and $ResMach_j$ indicates the number of residual machines after pruning by $q_t[1 : index_j]$. For those dimensions which are not in these pairs, we use linear interpolation to estimate their $ResMach$.

Here(fig?) is an example for the above procedure.

We use the following procedure to maintain this vector $EstResMach$:

4.6.2 Estimate the Transmission Cost

Once we have the vector $EstResMach$, we are ready to estimate the transmission cost *before* sending the query. For a query q, we could estimate its transmission cost in the first round

$$Cost_1 = TotalNumOfMach \times s_1 + Cost_{PRP}(TotalNumOfMach)$$

And for those round $j > 1$ as follows,

$$Cost_j = EstResMach[p_{j-1}] \times s_j + Cost_{PRP}(EstResMach[p_{j-1}])$$

where $p_j = \sum_{i=1}^{j} s_i$.

Give some explanation.

Therefore, we could solve this optimization problem to get the optimal pivots that could get the minimal total cost.

However, there are too many variables to decide in this problem. According to our experiments, the most crucial variable is the number of dimensions which will be sent in the first round, which is s_1 in this optimization problem. The other s_j don't have such huge influence like s_1. Therefore, we make all s_j be equal and then simplifiy this optimization problem as follows,

$$Cost_1 = TotalNumOfMach \times StartD + Cost_{PRP}(TotalNumOfMach)$$

And for those round $j > 1$,

$$Cost_j = EstResMach[p_{j-1}] \times EachLenD + Cost_{PRP}(EstResMach[p_{j-1}])$$

where $p_j = StartD + (j-1) \times EachLenD$.

Thus, the final optimization becomes as below,

$$\begin{aligned} \underset{StartD, EachLenD}{\text{minimize}} \quad & \sum_j Cost_j(StartD, EachLenD) \\ \text{subject to} \quad & StartD, EachLenD \in \mathbb{N} \end{aligned} \tag{4.11}$$

where $p_j = StartD + (j-1) \times EachLenD$.

4.6.3 Coordinate Descent to Decide the Pivots

Now we have reduced the number of variables to only two variables: $StartD$ and $EachLenD$. To solve this optimization problem efficiently, we apply the Coordinate Descent method as follows for each query.

algorthm?

After solving the optimal $StartD$ and $EachLenD$ for this query, we are able to decide its pivots as below:

$$p_j = StartD + (j-1) \times EachLenD \tag{4.12}$$

Since the vector $EstResMach$ would be updated for every new query, we will use Coordinate Descent to solve these pivots also for every new query.

4.7 Importance-Selecting Function and Overall Framework

At each round j for q_t, we need to decide what to send from P to each M_i and then calculate the bounds at M_i.

In this section, we describe the overview of our framework.

Chapter 5

Experiment

5.1 Experiment Setup

setup

5.1.1 Frameworks for Comparison

LeeWave, *LeeWave*, *Naive*,

CP, *PRP*, CP

5.1.2 Data Description

Put a table for summary. time, ann, f2 f3 mvd, trh,

Table 5.1: Summary for each dataset

Dataset	Feature	Num of Dimensions
ANN	SIFT	128
Flickr	Fea2	480
	Fea3	480
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Strikers	ST	Alan Smith
	ST	Mark Viduka

5.2 Comparison Among all Frameworks

$k = 10$, $|Q| = 500$, $M = 500, 1000, 1500, ...$ compare all diff frameworks for all data here. remember to add Mat cost.

5.3 Comparison Among our Framework with Different Configurations

There are many stages of algorithms which lead to the final version of our framework. Therefore, in this section, we would like to discuss the performance of our framwork with or without.

From **??** to our framework, we have enhanced it

5.3.1 Influence of the Orthogonal Transformation

NoW

5.3.2 Influence of the Threshold-Finding Procedure

NoPRP

5.3.3 Influence of the Coordinate Descent for Deciding Pivots

NoCD

5.4 Power of the Pruning Procedure

Comp among LeeWave, NoW, Main for ResSite.

Bibliography

[1] Mi-Yen Yeh, Kun-Lung Wu, Philip S. Yu, and Ming-Syan Chen. LeeWave: level-wise distribution of wavelet coefficients for processing $k$nn queries over distributed streams. *Proc. VLDB Endow.*, 2008.

[2] Zaiwen Wen and Wotao Yin. Optimization with orthogonality constraints. http://optman.blogs.rice.edu/, Aril 2012.

[3] Jui-Pin Wang, Yu-Chen Lu, Mi-Yen Yeh, Shou-De Lin, and Phillip B. Gibbons. Communication-efficient distributed multiple reference pattern matching for m2m systems. *Proc. of IEEE ICDM*, 2013.

[4] Apostolos N. Papadopoulos and Yannis Manolopoulos. Distributed processing of similarity queries. *Distributed and Parallel Databases*, 2001.