Special Talks for CSCI-UA.0480-006:

Statistical NLP: A Machine Learning Perspective

Miao Fan

Ph.D. candidate on C.S.

fanmiao.cslt.thu@gmail.com

0.1 Self-introduction

Miao Fan (Michael)
Bib: http://1drv.ms/1ijSyg9
Email: fanmiao.cslt.thu@gmail.com

4th year Ph.D. candidate on C.S. jointly supervised by
Tsinghua University (4 years) and **New York University (1 year).**

Research Interest: *Machine Learning* and *Natural Language Processing*.

Google Scholar: Just google "Miao Fan" and click the **1st** item.
https://scholar.google.com/citations?user=aPlHReAAAAAJ&hl=en

Hobby: table tennis, swimming, playing the piano.

0.2 Overview

- **What do we mainly learn from CSCI-UA.0480-006?**

  - NLP from the perspective of "**Linguistics**".

- **What I am and will be talking about in this special session are,**

  - NLP from the perspective of "**Statistics**" and "**Machine Learning**".

  - Some **canonical approaches** for **real-world applications**.

0.2 Overview

- ## How to define that a computer can learn (be intelligent)?
  - *A computer program* is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experiences **E**. (Tom Mitchell, CMU, 1997) [1].

- ## Real-world Applications?
  - Some tasks could be **hard coding (explicitly programmed).** Easy for machine but difficult for human beings:
    - **Calculator:** https://www.google.com/#q=calculator.
    - We know how to write commands (codes) to guide the machine process the task step by step.
  - Some are rather simple for human beings, but *hard for machine to process*:
    - **OK Google!**: https://www.google.com/ (Speech)
    - **Object recognition in image**: https://www.metamind.io/ (Image)
    - **Question Answering:** http://www.wolframalpha.com/ (Text)
    - **Product Recommendation**: http://nyti.ms/19LT8ic
    - We **DON'T** know how to write commands (codes) to guide the machine process the task step by step.
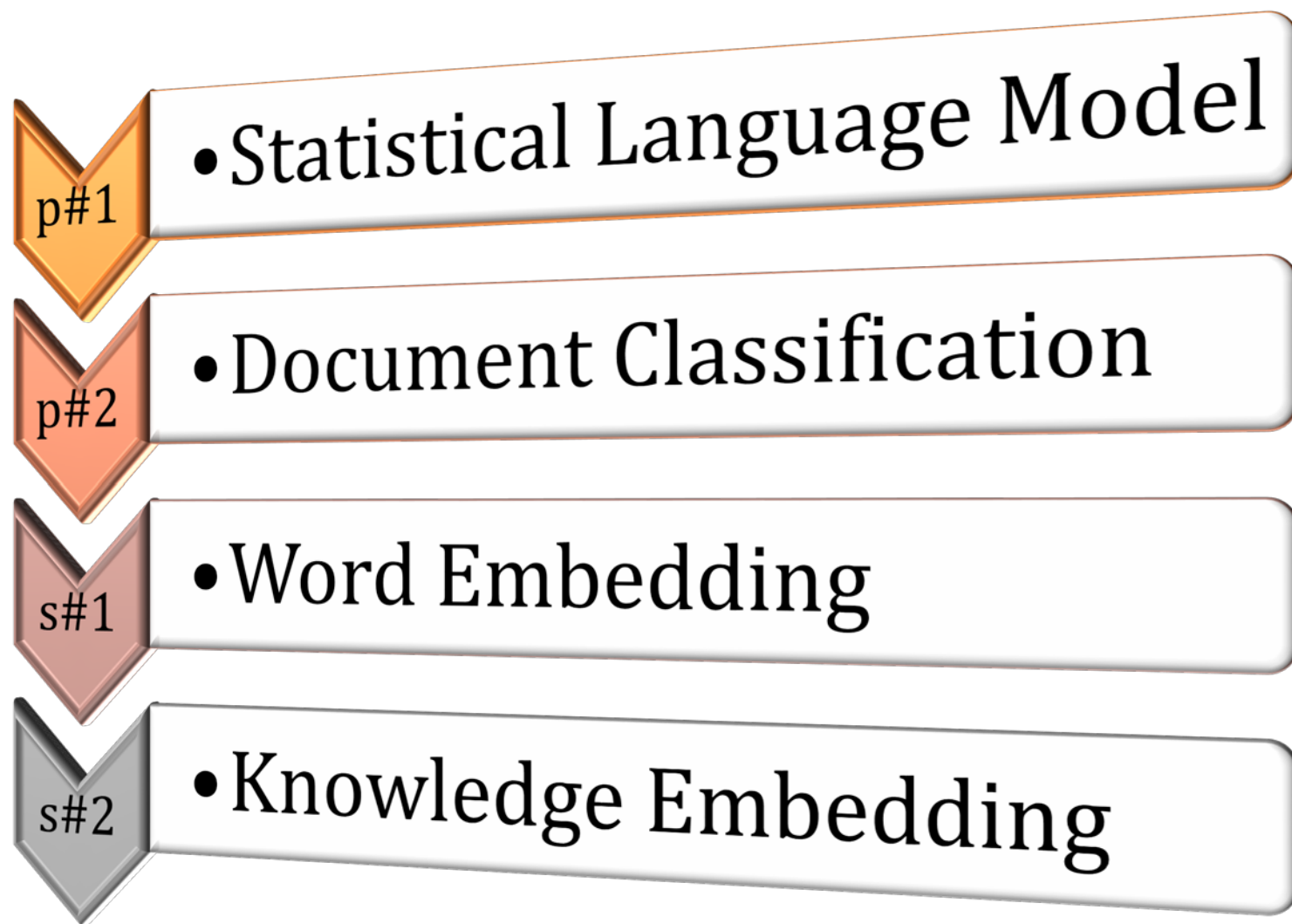
0.2 Overview

- Let's take an example about how to describe a real-world problem in Machine Learning definition: **Email Spamming**
  - Task T:
    - Filter spam emails
  - Experience E:
    - Emails labeled by "spam" or 'not spam'
  - Performance Measure P:
    - Accuracy?
    - Others?

0.3 Roadmap

- **Statistical Language Model** (p#1)
- **Document Classification** (p#2)
- **Word Embedding** (s#1)
- **Knowledge Embedding** (s#2)

Special Talks for CSCI-UA.0480-006:

Statistical NLP: A Machine Learning Perspective

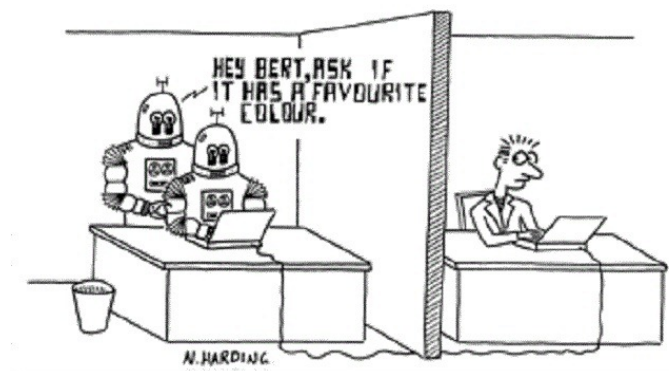### Precursor #1: Statistical Language Model (SLM)

P#1.1: Task (T) of Statistical Language Model

- It's rather flexible to define the task of SLM:
    - Make computer to **predict** (measure) whether a sentence is generated by a human.
    - Make computer to **generate** human language (sentence) automatically.

        For example,
        - *I am a student from NYU.*
        - *NYU a student from I am.*
        - Which one is more likely spoken by **an educated guy**? ☺

P#1.2: Experience (E) of Statistical Language Model

- We can *train* your computer to ***understand your tongue***.

  - Just feed the model with your daily spoken English.

  - The intelligent program is expected to ***improve*** the capability of understanding natural language ***better and better***, as we keep on feeding text corpus generated by human beings.

  - Start ***teaching*** your computer ***to write sentences***!

P#1.3: Statistical Language Model

- Let's regard sentences as words in sequence with STOP sign:

  - the dog barks STOP
  - the cat laughs STOP
  - the cat saw the dog STOP
  - the STOP
  - cat the dog the STOP
  - cat cat cat STOP
  - STOP

P#1.3: Statistical Language Model

- $Vocabulary\ Set\colon\ V\ =\ \{the, dog, laughs, saw, barks, cat\ ...\}$
- $A\ sentence\colon\ s = x_1 x_2 \ldots x_n; (x_i \in V)$
- We measure the probability of s:
  - $p(s)$
- For all possible expressions:
  - $\sum p(x_1, x_2, \ldots x_n) = 1$
- Let's recap:
    - *P(I am a student from NYU.) > P(NYU a student from I am.)*

P#1.3: Statistical Language Model

- *Ngram Model (Context):*
  - Bigram:

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1})$$

$$P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})$$

  - Trigram:

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$= \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

  - Unigram: Do it by yourself.

P#1.4: Performance (P) of Statistical Language Model

- How to measure the Capability of Understanding Language?
  - Perplexity!

- How is the perplexity (l) defined?
  - Suggesting that we have **m** sentences $(s^{(1)}, s^{(2)}, s^{(3)}, \ldots, s^{(m)})$ for testing.
  - And **M** to be the total number of words in the test corpus.

$$l = \frac{1}{M} \sum_{i=1}^{m} \log_2 p(s^{(i)})$$

P#1.5: Demo of Statistical Language Model with NLTK 3.0

- Therefore, we have to demo how to generating nGrams from your texts:

example_1.py

Note

The generate() method is not available in NLTK 3.0 but will be reinstated in a subsequent version.

Special Talks for CSCI-UA.0480-006:

Statistical NLP: A Machine Learning Perspective

Precursor #2: Document Classification (DC)

P#2.1: Task (T) of Document Classification

- Classify a document into a **pre-defined** category.
  - For example, New York Times

World   U.S.   Politics   N.Y.   Business   Opinion   Tech   Science   Health   Sports   Arts   Style   Food   Travel   Magazine   T Magazine   Real Estate   ALL

SPECIAL REPORT

How 4 Federal Lawyers Paved the Way to Kill Bin Laden

By CHARLIE SAVAGE 12:53 PM ET

The lawyers tried to prepare for any legal obstacles — and made it all but inevitable that Osama bin Laden would be killed, not captured.

151 Comments

The Opinion Pages

The Military Escalation in Iraq and Syria

By THE EDITORIAL BOARD

The United States is being drawn ever more deeply into a war that lacks an attainable end goal.

- Editorial: A Budget Deal to Live By, for Now
- Friedman: Telling Mideast Negotiators, 'Have a Nice Life'
- Edsall: Is There a Silver Lining

OP-ED CONTRIBUTORS

Why the Annual Mammogram Matters

By SUSAN R. DROSSMAN, ELISA R. PORT and EMILY B. SONNENBLICK

The American Cancer Society's new guidelines are wrong. Frequent testing is still the best.

- Room for Debate: Hurricane Sandy's Lessons for the Future
- Rattner: Don't Raise Interest Rates

P#2.2: Experiences (E) of Document Classification

- We feed **millions news about (Not about) politics** to intelligent machines.

Deeply Divided Republican Electorate Drifts Toward Ben Carson, Poll Shows

By JONATHAN MARTIN and MEGAN THEE-BRENAN

The survey of Republicans shows that Mr. Carson leading Donald J. Trump, who tied Jeb Bush at the presidential race, highlighting divisions within the party.

Monica Almeida/The New York Times

From left, Ben Carson, Donald J. Trump and Jeb Bush at the Republican presidential debate in September.

Politics

CNBC May Be the Big Winner of the Next Republican Debate

By JOHN KOBLIN

The cable financial news network is expecting to beat its own viewership records and is commanding top dollar for advertising as it airs Wednesday's Republican debate.

Ben Carson Puts Spotlight on Seventh-Day Adventists

By ALAN RAPPEPORT

The Republican presidential candidate's faith, little known to many, could prove to be both a strength and a liability as he moves forward.

N.F.L.'s Forays Into London Muddle Its Stance on Sports Betting

By KEN BELSON and JOE DRAPE
46 Minutes Ago

The N.F.L. has long opposed sports betting, but in Britain, where it has become legal and which now hosts N.F.L. games, the line has been blurred.

Tim Ireland/Associated Press

Fans pose for pictures at an N.F.L. fan rally in London before Sunday's of Sunday's game between the Bills and the Jaguars.

Not Politics

Thursday's Matchup: Miami Dolphins at New England Patriots
10 Minutes Ago

ON PRO BASKETBALL

Double Vision or Not, Derrick Rose Has His Sights Set on a Title
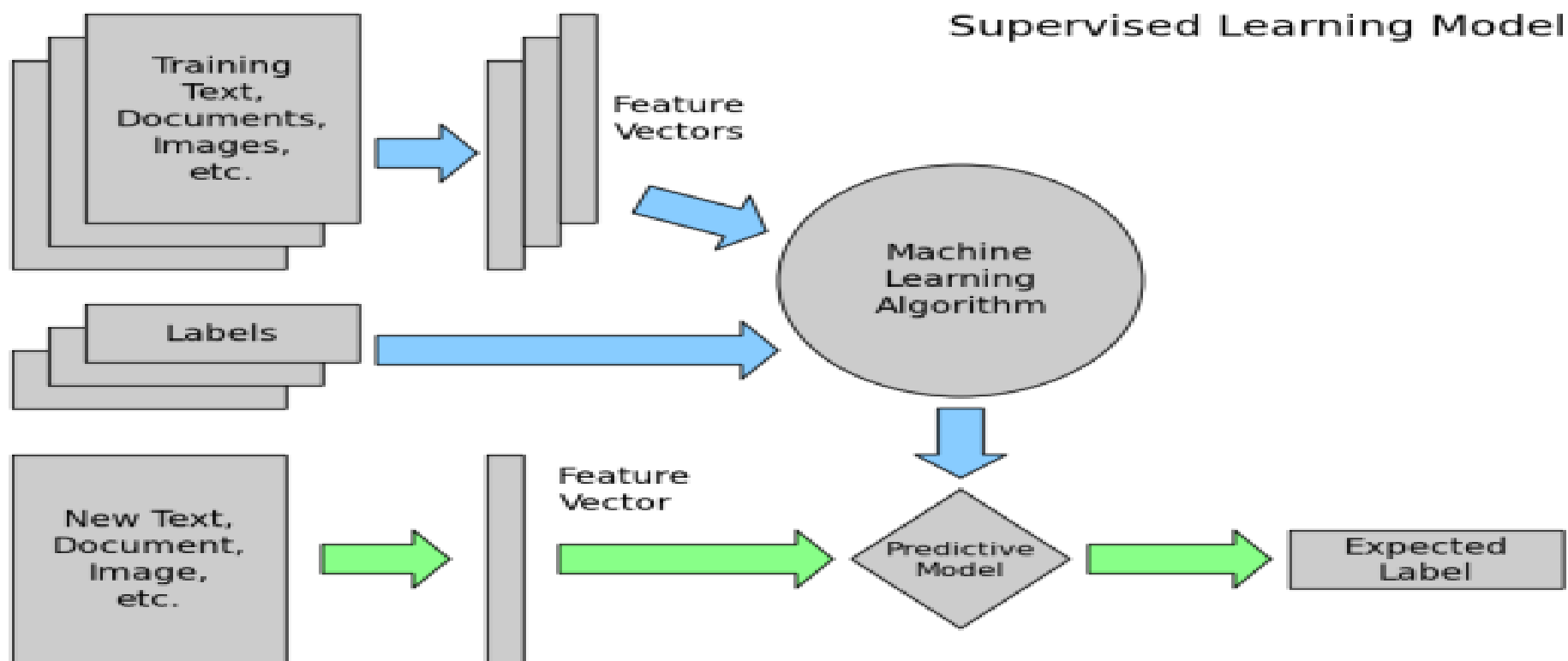
By HARVEY ARATON 12:26 PM ET

He was all the point guard rage a half-decade ago, but over the last four seasons, Rose was limited to 100 of a possible 312 games.

Dennis Wierzbicki/USA Today Sports, via Reuters

P#2.3: Document Classification Model

- Framework of **Supervised Learning**.

P#2.3: Document Classification Model

- Why we know this piece of news is mostly likely about politics?

By JONATHAN MARTIN and MEGAN THEE-BRENAN   OCT. 27, 2015

Email

Share

Tweet

Save

More

The latest New York Times-CBS News poll makes Republican Party divisions clear, from the choice of a presidential nominee to whether party members are willing to see their leaders compromise on legislation.

For the first time since The Times and CBS News began testing candidate preferences in July, the retired neurosurgeon Ben Carson has displaced Donald J. Trump as the leader of the large Republican field, although the difference is well within the poll's margin of sampling error. The churn in the field suggests more volatility as the contest draws closer to the primaries early next year.

Mr. Carson and Mr. Trump draw support from different segments of the Republican electorate.

From left, Ben Carson, Donald J. Trump, and Jeb Bush at the Republican presidential debate

P#2.3: Document Classification Model

- How could we know he is Mr. Trump, not Hillary?
  - Because of his keyword (China): https://www.youtube.com/watch?v=RDrfE9I8_hs
  - Because of his key phrase (Big League): http://www.slate.com/blogs/the_slatest/2015/09/24/bigly_or_big_league_what_exactly_is_donald_trump_saying.html
  - If the computer know that P(China|Trump) > P(China|Other candidates) from news.
  - Then given China, P(Trump|China)?

P#2.3: Document Classification Model

- *1)* **Naïve Bayes Model:**
  - *We'd like to know $P(y|x)$, given a document $x = (x_1, x_2, x_3, \ldots, x_n)$.* (*Feature Vectors*)
  - *y is the variable of categories.* (*Labels*)

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Naive Bayes Assumption: $P(x_i \mid y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i \mid y),$

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

P#2.3: Document Classification Model

- *2) Logistic Regression Model:*
  - *Given a document* $x = (x_1, x_2, x_3, \ldots, x_n)$. *(Feature Vectors)*
  - *y is the variable of categories.* *(Labels)*

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

$\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ is the parameter vector corresponding to $x = (x_1, x_2, x_3, \ldots, x_n)$

P#2.3: Document Classification Model

- **How do we generate the *feature vectors* from documents?**

-   Bag of words (BOW) Binary representation:

some	tigers	live	in	the	zoo	green	is	a	color	go	to	new	york	city	
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	class 1
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	class 2
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	class 3

< 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0 > Unknown class

< 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 > class 1 (6 matching terms)
< 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0 > class 2 (14 matching terms - winner!!)
< 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 > class 3 (7 matching terms)

-   Any Other representation? (TFIDF?)

some	tigers	live	in	the	zoo	green	is	a	color	go	to	new	york	city	
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	unknown class

P#2.4: Performance (P) of Document Classification

- How to measure the performance of (Binary) classification?
  - We have multiple metrics:

		True condition			
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma\ \text{Condition positive}}{\Sigma\ \text{Total population}}$	
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma\ \text{True positive}}{\Sigma\ \text{Test outcome positive}}$	False discovery rate (FDR) $= \frac{\Sigma\ \text{False positive}}{\Sigma\ \text{Test outcome positive}}$
	Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) $= \frac{\Sigma\ \text{False negative}}{\Sigma\ \text{Test outcome negative}}$	Negative predictive value (NPV) $= \frac{\Sigma\ \text{True negative}}{\Sigma\ \text{Test outcome negative}}$
	Accuracy (ACC) = $\frac{\Sigma\ \text{True positive} + \Sigma\ \text{True negative}}{\Sigma\ \text{Total population}}$	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma\ \text{True positive}}{\Sigma\ \text{Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\Sigma\ \text{False positive}}{\Sigma\ \text{Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR-}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma\ \text{False negative}}{\Sigma\ \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma\ \text{True negative}}{\Sigma\ \text{Condition negative}}$	Negative likelihood ratio (LR-) $= \frac{\text{FNR}}{\text{TNR}}$	

P#2.4: Performance (P) of Document Classification

- Besides the *Accuracy*, in some cases, we care more about ***Precision & Recall***.
    - Suppose that we have a **classification task** that ***predict*** whether a tumor is **malignant or benign**, based on some *medical features* such as the tumor's size, position, etc.
    - Suppose that the model we design will label **malignant tumors as positive.**
    - If we predict a tumor as positive (malignant), and the truth is: it is **malignant**! **Then True Positive (TP).** Congratulations!
    - If we predict a tumor as negative (benign), but the truth is: it is **malignant**! **OMG! That's False Negative(FN). That kills people.**
    - **Therefore, we care more on recall (The proportion of malignant tumor that we need to find), rather than precision**.

True positive	False positive (Type I error)
False negative (Type II error)	True negative

P#2.5: Demo of Document Classification with Python

1. Structured document from library:

https://www.kaggle.com/c/titanic
http://localhost:8888/notebooks/PyNotebook/example_2.ipynb#

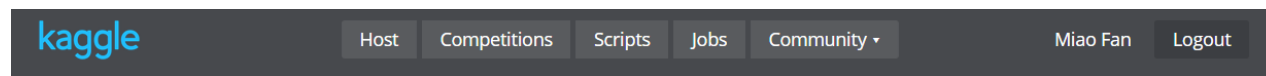How to choose features?
What kind of classifier do we use?

You might use **Scikit-learn** (Machine Learning Modules), **Pandas** (Data Manipulation Package) and *NLTK* (NLP toolkits).

I also suggest you to learn this series of video:

P#2.5: Demo of Document Classification with Python

2. Unstructured Web document

https://www.kaggle.com/c/dato-native

Leave it to YOU!

You may also need Spark! (Distributed Computing)

Special Talks for CSCI-UA.0480-006:

Statistical NLP: A Machine Learning Perspective

State-of-the-art Approach #1: Word Embedding (WE)

S#1.1 Preliminary

- Let's recap:

  - We've talked about "Statistical Language Modeling".

  - Given "*The cat is walking in the bedroom.*"; "*The cat is running across the street!*"

  - Basically, Pr('cat'|'the') = $\frac{\#(cat,the)}{\#(the)}$.

S#1.2 Motivation

- For example, given "*The cat is walking in the bedroom.*" in the training corpus,
  - Could we generalize to make the sentence *"A dog was running in a room."*
  - *It seems impossible mission for Statistical Language Model based on Ngram learnt by MLE.*
  - *However, we could find word similarity between:*
    - *The|A*
    - *cat|dog*
    - *is/was*
    - *walking/running*
    - *bedroom/room*

S#1.2 Motivation

- ## The curse of dimensionality!

- *If we have a corpus which contains 1,000 sentences (Not many), 5000 tokens (5 tokens per sentence), 2000 words (size of vocabulary) .*

- *How many possible BI-GRAM terms we need to train? 2000 ^ 2 = ?*

- *How many words daily used in English?*

  - *http://www.lingholic.com/how-many-words-do-i-need-to-know-the-955-rule-in-language-learning-part-2/*

LANGUAGE	LARGEST DICTIONARY	NUMBER OF WORDS
Chinese	汉语大词典 (Hanyu Da Cidian. Lit: Comprehensive Chinese Word Dictionary)	370,000 words; 23,000 head Chinese character entries
English	The Second Edition of the 20-volume *Oxford English Dictionary*	171,476 words in current use, and 47,156 obsolete words; 615,100 definitions

- *Even bi-gram need to train 170,000 ^ 2 = ?*

- ***Every float is 4 bytes, 4 bytes * 170,000 ^ 2 = 115GB Memory!***

S#1.2 Motivation

- Let's recap:
  - How we represent features? BOW

  < 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0 > Unknown class

  < 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 > class 1 (6 matching terms)
  < 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0 > class 2 (14 matching terms - winner!!)
  < 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 > class 3 (7 matching terms)

  - Rather sparse, difficult to calculate the similarity with COSINE?
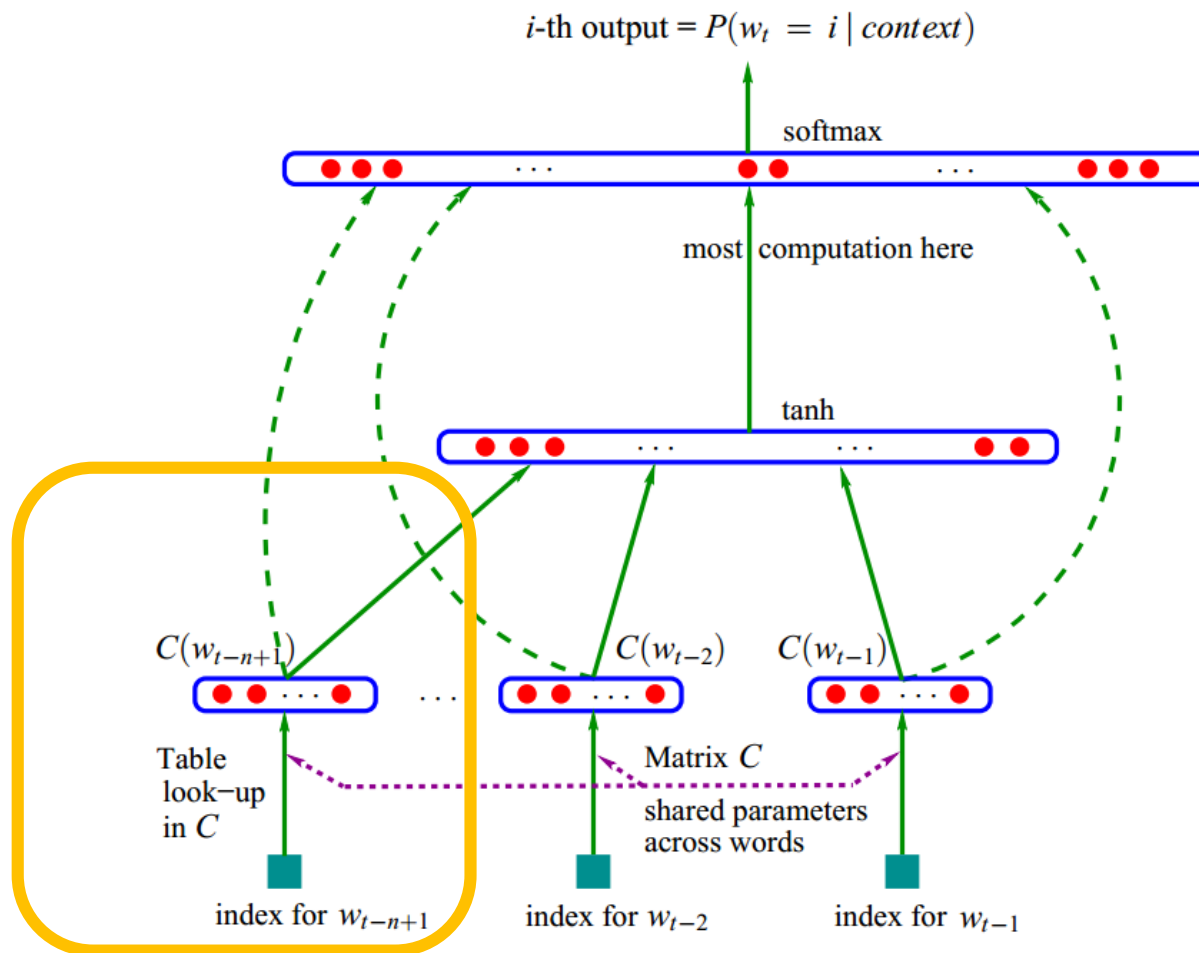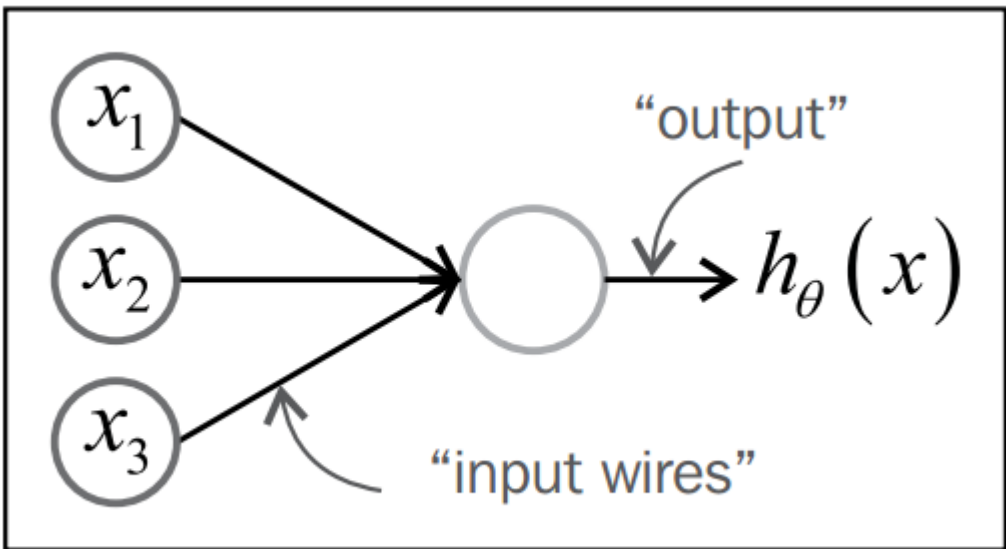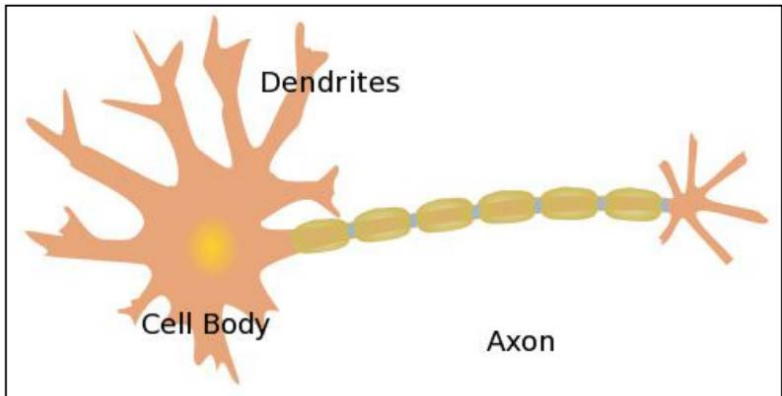 - What if Cat = (0.6,0.8), dog = (0.7, 0.6)? More dense, similar!

S#1.2 Motivation

Figure from [3]: http://www.socher.org/

S#1.3 Model

i-th output $= P(w_t = i \mid context)$

softmax

most computation here

tanh

$C(w_{t-n+1})$ $C(w_{t-2})$ $C(w_{t-1})$

Table
look–up
in C

Matrix C
shared parameters
across words

index for w_{t-n+1} index for w_{t-2} index for w_{t-1}

Figure from [2]: Neural Language Model

S#1.3 Model

$C(w_{t-n+1})$

Table look-up in C

index for w_{t-n+1}

Dendrites

Cell Body

Axon

x_1

x_2

x_3

"output"

$h_\theta(x)$

"input wires"
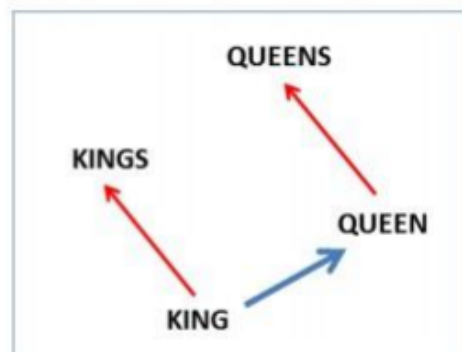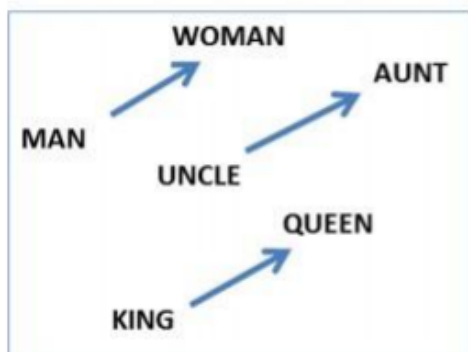
Do you still remember the "Logistic Regression" Model?

S#1.4: Discovery

- Promising discovery in **Word Embedding, in which each word is represented by a low-dimensional vector. Ex. King = (0.6, 0.24, 0.4, …, 0.3);**

Measuring Linguistic Regularity

- Syntactic/Semetic Test

$$C(king) - C(queen) \approx C(man) - C(woman)$$

$$C(king) - C(man) + C(woman) \approx C(queen)$$

These representations are surprisingly good at capturing syntactic and semantic regularities in language, and that each relationship is characterized by a relation-specific vector offset.

Figure from [4]: Neural Language Model

S#1.5: Low-dimensional representation

- We usually encode each word into a **K (K = 50, 100 or 200)** dimensional vector space.

How many words daily used in English?

http://www.lingholic.com/how-many-words-do-i-need-to-know-the-955-rule-in-language-learning-part-2/

LANGUAGE	LARGEST DICTIONARY	NUMBER OF WORDS
Chinese	汉语大词典 (Hanyu Da Cidian. Lit: Comprehensive Chinese Word Dictionary)	370,000 words; 23,000 head Chinese character entries
English	The Second Edition of the 20-volume *Oxford English Dictionary*	171,476 words in current use, and 47,156 obsolete words; 615,100 definitions

How much memory space do you need?
*Every float is 4 bytes, 4 bytes * 170,000 * 200 = 136MB Memory!*
Compared with 115GB! (1000 TIMES)@

Special Talks for CSCI-UA.0480-006:

Statistical NLP: A Machine Learning Perspective

State-of-the-art Approach #2: Knowledge Embedding (KBE)

S#2.1: Preliminary

- **1. What is knowledge?**
  - We distill the explosive **unstructured web texts** into **structured tables** which record the facts of the world.
  - For example, **Jinping Xi** is the chairman of CCP.

- **2 How we present or store the knowledge?**
  - For now, we present or store the knowledge in triplets,

    *i.e. (head_entity, relationship, tail_entity), abbreviated as **(h, r, t)**.*
  - For example, (Jinping Xi, chairman of, CCP)

- **3 Is there any free-access repositories of knowledge?**
  - Of course, you can freely download the whole Knowledge Base online.
  - For example, Freebase, NELL, Yago, WordNet…
  - **Just Google THEM!**
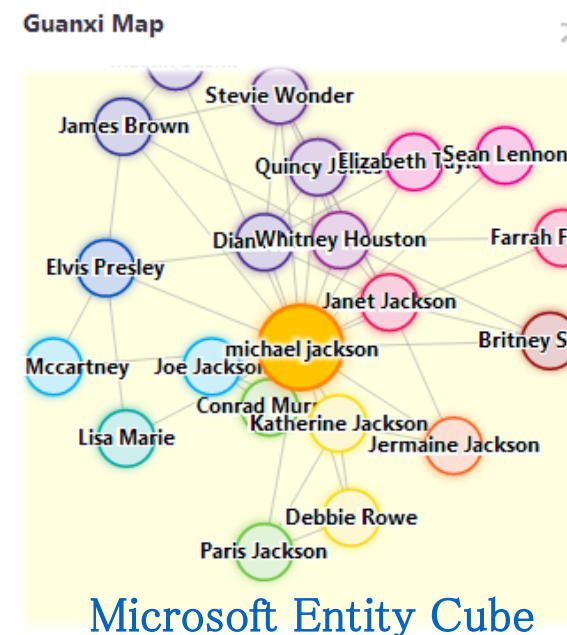
S#2.1: Preliminary

- **4 Is knowledge base really useful?**
  - Sure it is. Applications such as **Google Knowledge Graph** and **Microsoft Entity Cube.** We discover the connections between entities around the world.
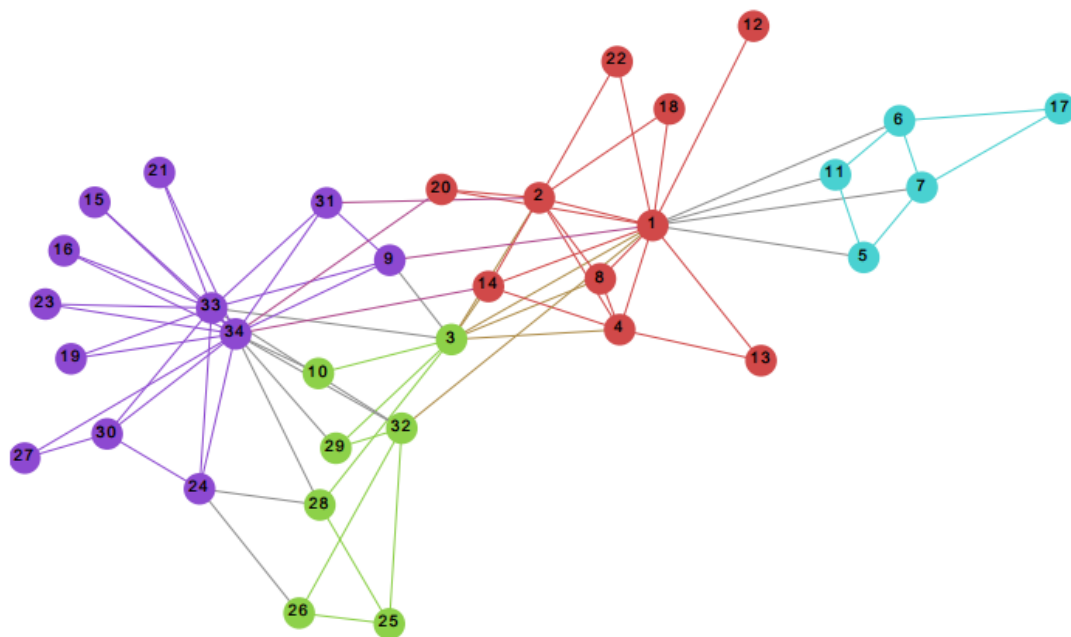

Google Knowledge Graph

Microsoft Entity Cube
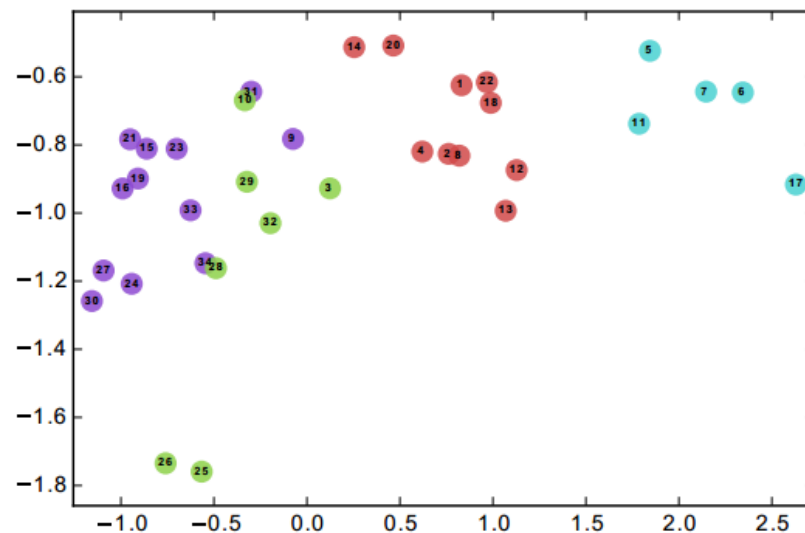
S#2.2: Motivation

- However, the KBs we have are **far from completion**.
  - Recent Study on **Freebase by Google Research (WWW 2014)** shows that **71% PERSONS have no known place of birth**, **94%** have no known **parents**, and **99%** have no known **ethnicity**.

- Therefore, we need to explore methods on **automatically** completing knowledge base. (The task: T)

- Here, we focus on knowledge **self-inference** without extra text corpus.
  - **A simple rule for relation inference:**
    - 1st Triplet : (Miao Fan, born in, Liaoning)
    - 2nd Triplet: (Liaoning, province of, China).
    - => rule inference, **new fact: (Miao Fan, Nationality, Chinese).**
  - **But the question is: is it possible to heuristically design rules that adequate for billions of facts?**
    - **Tough work!!**

S#2.2: Motivation

(a) Input: Karate Graph   (b) Output: Representation
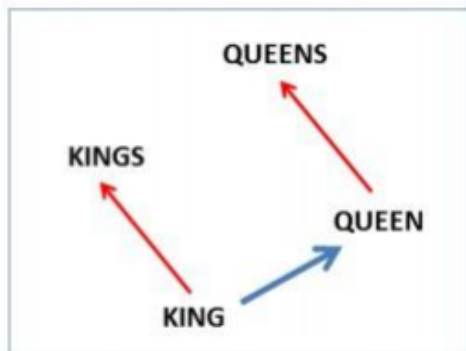
Figure from [5]: DeepWalk

S#2.2: Motivation

- Promising discovery in **Word Embedding, in which each word is represented by a low-dimensional vector. Ex. King = (0.6, 0.24, 0.4, ..., 0.3);**

Measuring Linguistic Regularity
- Syntactic/Semetic Test

$$C(king)-C(queen) \approx C(man)-C(woman)$$

$$C(king)-C(man)+C(woman) \approx C(queen)$$

These representations are surprisingly good at capturing syntactic and semantic regularities in language, and that each relationship is characterized by a relation-specific vector offset.

S#2.2: Motivation

- How about **Knowledge Embedding**?

In the Word Embedding Space:
$$China - Beijing \approx France - Paris$$

How about Knowledge Embedding Space?
$$China - Beijing \approx capital\_city\_of$$

Therefore, given a triplet (h, r, t),
$$h + r \approx t$$

(h:Beijing, r:**capital_city_of**, t: **China**)

(h:Paris, r: **capital_city_of**, t: **France**

S#2.3: Modeling

Fig. 1. **LMNNE** has two objects optimized simultaneously: *PULL* the positive tail entities (t^+) close to $\mathbf{h} + \mathbf{r}$ and *PUSH* the negative tail entities (t^-) out of the margin γ. In this case, all the entities and relations are embedded into the 2D vector space and we use L_2 *norm* to measure the distance.

Figure from [6]: LMNNE

S#2.3: Modeling

- **Triplet measurement:**

$$f_r(h, t) = ||h + r - t||, \qquad (1)$$

- **Pull** the positive triplets **Together!**

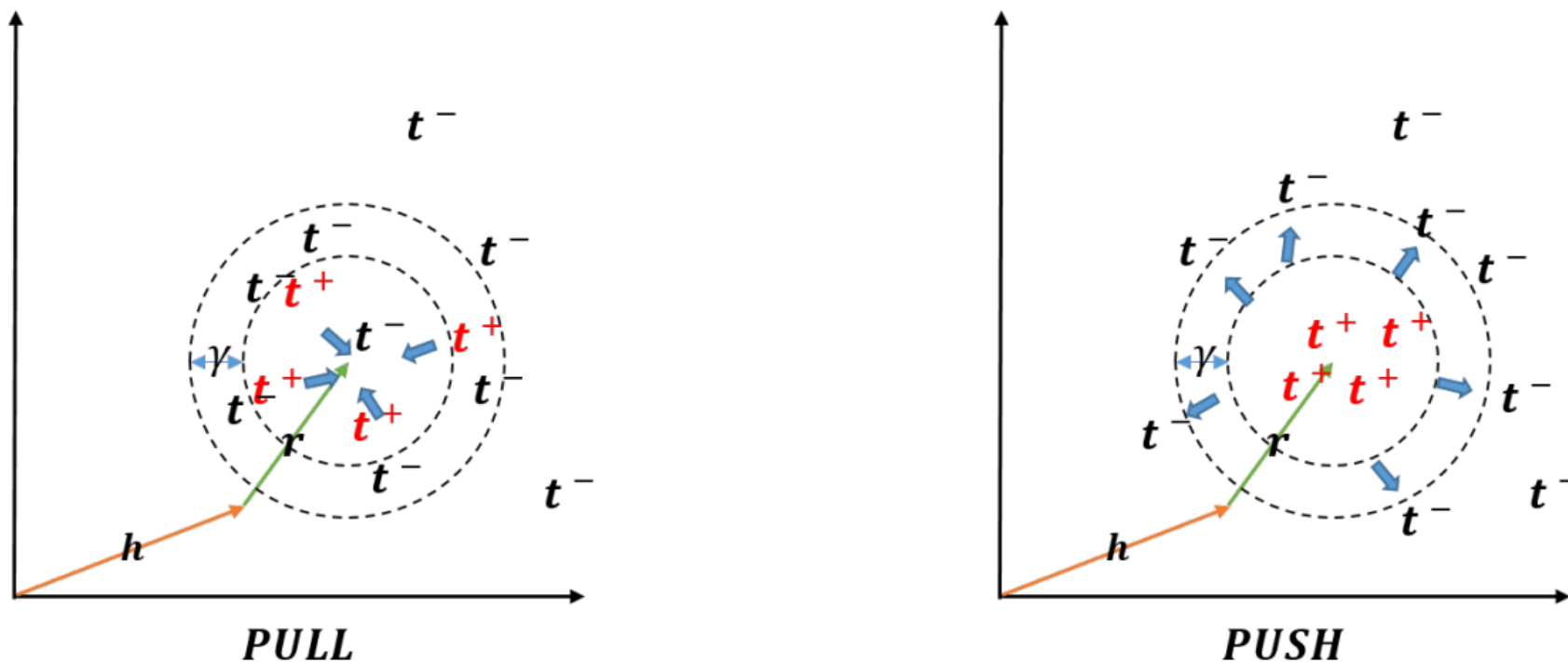$$\mathcal{L}_{pull} = \text{Min} \sum_{(h,r,t)\in\Delta} \sum_{(h^+,r,t^+)\in\Delta^+_{(h,r,t)}} (||h - h^+|| + ||t - t^+||).$$

$$(2)$$

- **Push** the negative triplets **Away!**

$$\mathcal{L}_{push} = \text{Min} \sum_{(h,r,t)\in\Delta} \sum_{(h^-,r,t^-)\in\Delta^-_{(h,r,t)}} [\gamma + f_r(h,t) - f_r(h^-,t^-)]_+$$

$$(3)$$

- **Overall Objective:**

$$\mathcal{L} = \text{Min} \quad \mu\mathcal{L}_{pull} + (1 - \mu)\mathcal{L}_{push}. \qquad (4)$$

S#2.4: Algorithms

Algorithm 1   The Learning Algorithm of **LMNNE**

Input:

Training set $\Delta = \{(h, r, t)\}$, entity set E, relation set R; dimension of embeddings d, margin γ, learning rate α and β for $\mathcal{L}_{pull}$ and $\mathcal{L}_{push}$ respectively, convergence threshold ϵ, maximum epochs n and the trade-off μ.

1: **foreach** $\mathbf{r} \in R$ **do**
2:     $\mathbf{r} := \text{Uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
3:     $\mathbf{r} := \frac{\mathbf{r}}{|\mathbf{r}|}$
4: **end foreach**
5: $i := 0$
6: **while** $Rel.loss > \epsilon$ and $i < n$ **do**
7:     **foreach** $\mathbf{e} \in E$ **do**
8:        $\mathbf{e} := \text{Uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
9:        $\mathbf{e} := \frac{\mathbf{e}}{|\mathbf{e}|}$
10:     **end foreach**
11:     **foreach** $(h, r, t) \in \Delta$ **do**
12:        $(h', r, t') := \text{Sampling}(\Delta'_{(h,r,t)})$
13:        **if** $(h', r, t') \in \Delta^+_{(h,r,t)}$ **then**
14:           Updating: $\nabla_{(h,r,t,h',t')}\mathcal{L}_{pull}$ with: $\alpha\mu$
15:        **end if**
16:        **if** $(h', r, t') \in \Delta^-_{(h,r,t)}$ **then**
17:           Updating : $\nabla_{(h,r,t,h',t')}\mathcal{L}_{push}$ with: $\beta(1-\mu)$
18:        **end if**
19:     **end foreach**
20: **end while**

Output:

All the embeddings of e and r, where $e \in E$ and $r \in R$.

Step 1: Uniformly Initial Vectors

Step 2: Pull positive triplets

Step 3: Push negative triplets

Step 4: Updating Embeddings with SGD

S#2.5: Experiments

TABLE I.   STATISTICS OF THE DATASETS USED FOR LINK PREDICTION
TASK.

DATASET	WN18	FB15K
#(ENTITIES)	40,943	14,951
#(RELATIONS)	18	1,345
#(TRAINING EX.)	141,442	483,142
#(VALIDATING EX.)	5,000	50,000
#(TESTING EX.)	5,000	59,071

TABLE II.   LINK PREDICTION RESULTS. WE COMPARED OUR PROPOSED LMNNE WITH THE STATE-OF-THE-ART METHOD (TRANSE) AND OTHER
PRIOR ARTS.

DATASET	WN18				FB15K			
METRIC	MEAN RANK		MEAN HIT@10		MEAN RANK		MEAN HIT@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
Unstructured	315	304	35.3%	38.2%	1,074	979	4.5%	6.3%
RESCAL	1,180	1,163	37.2%	52.8%	828	683	28.4%	44.1%
SE	1,011	985	68.5%	80.5%	273	162	28.8%	39.8%
SME (LINEAR)	545	533	65.1%	74.1%	274	154	30.7%	40.8%
SME (BILINEAR)	526	509	54.7%	61.3%	284	158	31.3%	41.3%
LFM	469	456	71.4%	81.6%	283	164	26.0%	33.1%
TransE	294.4	283.2	70.4%	80.2%	243.3	139.9	36.7%	44.3%
LMNNE	257.3	245.4	73.7%	84.1%	221.2	107.4	38.3%	48.2%

$Given\ h, r, predicting\ t(s)$

$RANK \rightarrow d(h + r - t)$

1. Link Prediction (predict t, give h and r)
2. Triplet Classification.

(Jinping Xi, chairman of, ?)

S#2.6: Conclusion

- Contributions:

  - From *sparse representations* to *dense representations*.
    - Low-dimensional vector spaces

  - Facilitate *statistical learning*.
    - Similarity & probability computing.

  - *Scalability* possible.
    - Make it possible to tackle with large-scale graph computing.

S#2.7: Future Work

- Several promising directions if you would like to follow our work:

    - **Knowledge Embedding with text corpus.**
        - **How about adopt Wikipedia. Please Check Miao Fan's Google Scholar**

    - **Parallel SGD Training for Knowledge Embedding.**
        - **Map-reduce. Please Check Miao Fan's Google Scholar.**

    - **Question-Answering Embedding?**
        - Rank(Q(What's the capital city of China) · A(Beijing))?

Acknowledgments

- Thanks to the **Instructor:** Adam Meyers for his comments.

- Thanks to Prof. Ralph Grishman and all the members of **Proteus Project**.

References

- [1] Machine Learning, Tom Mitchell, McGraw Hill, 1997.
- [2] Bengio, Yoshua, et al. "A neural probabilistic language model." *The Journal of Machine Learning Research* 3 (2003): 1137-1155.
- [3] Huang, Eric H., et al. "Improving word representations via global context and multiple word prototypes." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012.
- [4] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." In *HLT-NAACL*, pp. 746-751. 2013.
- [5] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 701-710). ACM.
- [6] Fan, Miao, Qiang Zhou, Thomas Fang Zheng, and Ralph Grishman. "Large Margin Nearest Neighbor Embedding for Knowledge Representation." *arXiv preprint arXiv:1504.01684* (2015).

Thanks for your attention!

Stay Hungry, Stay Foolish.
fanmiao.cslt.thu@gmail.com