Microsoft Research
微软亚洲研究院

THE UNIVERSITY of WISCONSIN MADISON

ACL 2008: Semi-supervised Learning Tutorial

John Blitzer and Xiaojin Zhu

http://ssl-acl08.wikidot.com

What is semi-supervised learning (SSL)?

- Labeled data (entity classification)

  - …, says Mr. **Cooper**, vice president of …
  - … **Firing Line** Inc., a **Philadelphia** gun shop.

Labels

person
location
organization

- Lots more unlabeled data

  - …, Yahoo's own Jerry Yang is right …
  - … The details of Obama's San Francisco mis-adventure …

Can we build a better model from both labeled and unlabeled data?

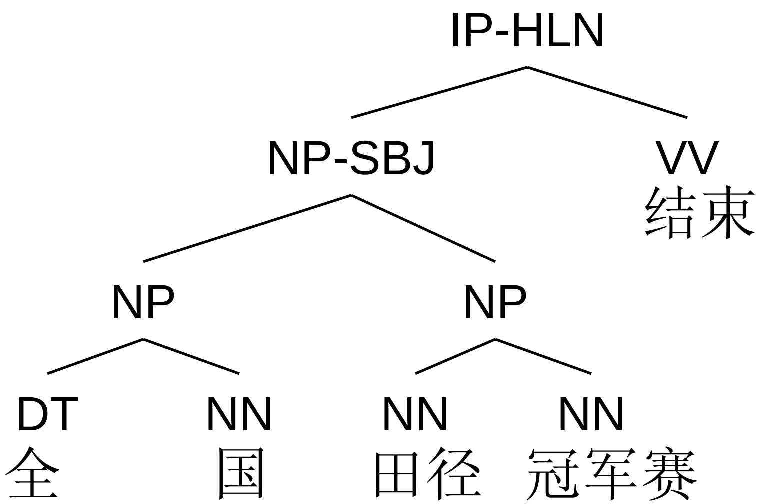
Who else has worked on SSL?

- **Canonical NLP problems**
  - **Tagging** (Haghighi and Klein 2006)
  - **Chunking, NER** (Ando & Zhang 2005)
  - **Parsing** (McClosky & Charniak 2006)

- **Outside the classic NLP canon**
  - **Entity-attribute extraction** (Bellare et al. 2007)
  - **Sentiment analysis** (Goldberg & Zhu 2006)
  - **Link spam detection** (Zhou et al. 2007)
  - **Your problem?**

Anti-SSL arguments: practice

- **If a problem is important, we'll find the time / money / linguists to label more data**



The national track & field championships concluded

Penn Chinese Treebank

2 years to annotate 4000 sentences

I want to parse the baidu zhidao question-answer database.

Who's going to annotate it for me?

Anti-SSL arguments: theory

- **“But Tom Cover said”: (Castelli & Cover 1996)**
  - Under a specific generative model, labeled samples are exponentially more useful than unlabeled

- **The semi-supervised models in this tutorial make different assumptions than C&C (1996)**

- **Today we’ll also discuss new, positive theoretical results in semi-supervised learning**

Why semi-supervised learning?

- **I have a good idea, but I can't afford to label lots of data!**

- **I have lots of labeled data, but I have even more unlabeled data**
  - **SSL: It's not just for small amounts of labeled data anymore!**

- **Domain adaptation:** I have labeled data from 1 domain, but I want a model for a different domain

Goals of this tutorial

1) **Cover the most common classes of semi-supervised learning algorithms**

2) **For each major class, give examples of where it has been used for NLP**

3) **Give you the ability to know which type of algorithm is right for your problem**

4) **Suggest advice for avoiding pitfalls in semi-supervised learning**

Overview

1) **Bootstrapping (50 minutes)**
   - **Co-training**
   - **Latent variables with linguistic side information**

2) **Graph-regularization (45 minutes)**

3) **Structural learning (55 minutes)**
   - **Entity recognition, domain adaptation, and theoretical analysis**

Some notation

labeled instances are pairs $(\mathbf{x}, y)$

learners or hypotheses $h, f : \mathbf{x} \rightarrow y$

labeled data $\{(\mathbf{x}, y)_i\}_{i=1}^{\ell}$

unlabeled data $\{\mathbf{x}_i\}_{i=\ell+1}^{m+\ell}$ available at train time

test data $\{(\mathbf{x}, y)\}$ unavailable at train time

Bootstrapping: outline

- The general bootstrapping procedure
- Co-training and co-boosting
- Applications to entity classification and entity-attribute extraction
- SSL with latent variables, prototype learning and applications

Bootstrapping

- On labeled data, minimize error
- On unlabeled data, minimize a proxy for error derived from the current model

- Most semi-supervised learning models in NLP

1) Train model on labeled data
2) Repeat until converged
   a) Label unlabeled data with current model
   b) Retrain model on unlabeled data

Back to named entities

- Naïve Bayes model $p(y|\mathbf{x}) \propto p(y)\ \Pi_i\ p(x_i|y)$

Features: <left word = “Mr.”>

Label: “Person”

- Parameters estimated from counts $c(x_i, y)$

Bootstrapping step	**Data**	**Update action**
Estimate parameters	Says Mr. **Cooper**, vice president	$c(\text{LW=Mr. , Person})\text{++}$
Label unlabeled data	Mr. **Balmer** has already faxed	Label **Balmer** “Person”
Retrain model	Mr. **Balmer** has already faxed	$c(\text{MW=Balmer , Person})\text{++}$ $c(\text{LW=Mr. , Person})\text{++}$

Bootstrapping folk wisdom

- **Bootstrapping works better for generative models than for discriminative models**
  - **Discriminative models can overfit some features**
  - **Generative models are forced to assign probability mass to all features with some count** $c(x_i, y)$

- **Bootstrapping works better when the naïve Bayes assumption is stronger**
  - **“Mr.” is not predictive of “Balmer” if we know the entity is a person** $p(x_i, x_j|y) = p(x_i|y)p(x_j|y)$

Two views and co-training

- **Make bootstrapping folk wisdom explicit**
  - There are two views of a problem.
  - Assume each view is sufficient to do good classification
- **Named Entity Classification (NEC)**
  - 2 views: Context vs. Content
  - says Mr. Cooper, a vice president of . . .

General co-training procedure

- **On labeled data, maximize accuracy**

- **On unlabeled data, constrain models from different views to agree with one another**

- **With multiple views, any supervised learning algorithm can be co-trained**

Co-boosting for named entity classification

Collins and Singer (1999)

- **A brief review of supervised boosting**
  - Boosting runs for t=1…T rounds.
  - On round t, we choose a base model $h_t(\mathbf{x})$ and weight α_t
  - For NLP, the model at round t, $h_t(\mathbf{x})$ identifies the presence of a particular feature and guesses or abstains

$$h^i(\mathbf{x}) = \begin{cases} \pm 1, & x_i = 1 \\ 0, & \text{otw.} \end{cases}$$

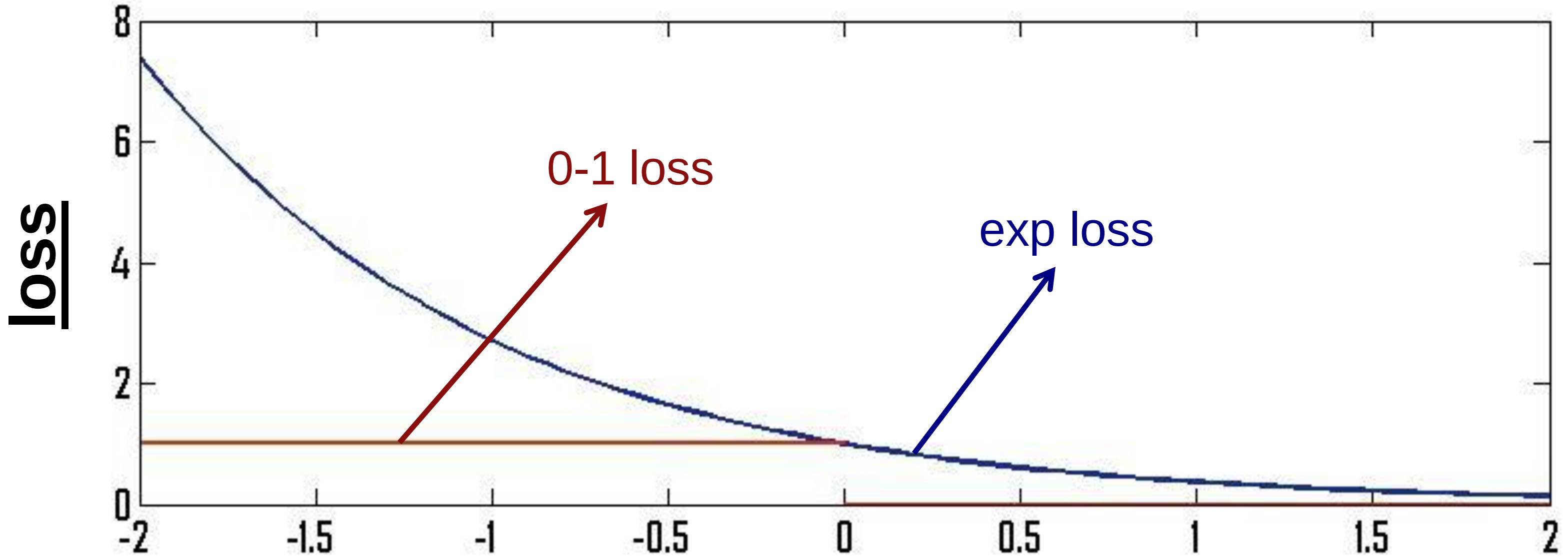
  - Final model: $f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

Boosting objective

Normal boosting: At each round t, we set α_t and $h_t(\mathbf{x})$ to minimize

Current model, steps 1. . .t-1

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\exp\left(-y_i\left(\sum_{s=0}^{t-1}\alpha_s h_s(\mathbf{x}_i)+\alpha_t h_t(\mathbf{x}_i)\right)\right)$$



Co-boosting objective

Let $f^1(\mathbf{x}_i^1)$, $f^2(\mathbf{x}_i^2)$ be the boosted classifiers from views 1 and 2, respectively. Then the co-boost loss for round t is:

view 2 loss

subscript: round of boosting

superscript: view

$$
\begin{aligned}
&\text{trainloss} \\
&+ \frac{1}{m} \sum_{i=\ell}^{m+\ell} \exp\left(-f(\mathbf{x}_i^1) \left(\sum_{s=0}^{t-1} \alpha_s h_s^2(\mathbf{x}_i^2) + \alpha_t h_t^2(\mathbf{x}_i^2) \right) \right) \\
&+ \frac{1}{m} \sum_{i=\ell}^{m+\ell} \exp\left(-f(\mathbf{x}_i^2) \left(\sum_{s=0}^{t-1} \alpha_s h_s^1(\mathbf{x}_i^1) + \alpha_t h_t^1(\mathbf{x}_i^1) \right) \right)
\end{aligned}
$$

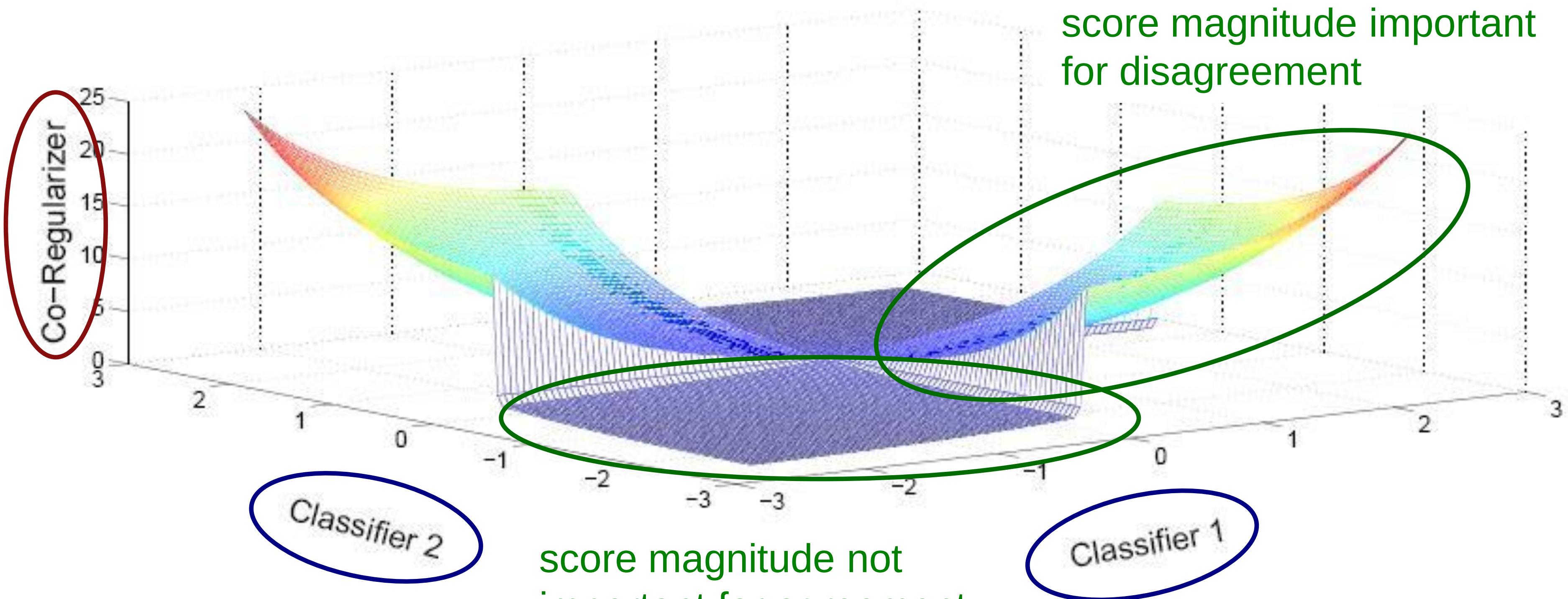
view 1 loss

Unlabeled co-regularizer

Scores of individual ensembles (x- and y-axis) vs.

Co-regularizer term (z-axis)

score magnitude important for disagreement



score magnitude not important for agreement

Co-boosting updates

- Optimize each view separately.
  - Set hypothesis h_t^2, α_t^2 to minimize

$$\frac{1}{m}\sum_{i=\ell+1}^{m+\ell} \exp\left(-f(\mathbf{x}_i^1)\left(\sum_{s=0}^{t-1}\alpha_s h_s^2(\mathbf{x}_i^2) + \alpha_t h_t^2(\mathbf{x}_i^2)\right)\right)$$

  - Similarly for view 1

- Each greedy update is guaranteed to decrease one view of the objective

Basic co-boosting walk-through

Labeled: Mr. **Balmer** has already faxed

Unlabeled: says Mr. **Smith**, vice president of

Adam Smith wrote "The Wealth of Nations"

Co-boosting step	Data	Update action
Update context view	**Mr.** Balmer has already faxed	$h_1^1(\mathbf{x}) = I\left(\text{Mr.} \in \mathbf{x}_1\right)$
Label unlabeled data	says Mr. **Smith**, vice president	Label "Person"
Update content view	says Mr. **Smith**, vice president	$h_1^2(\mathbf{x}) = I\left(\text{Smith} \in \mathbf{x}_2\right)$
Label unlabeled data	**Adam Smith** wrote "The Wealth of Nations"	Label "Person"
Update context view	Adam Smith **wrote** . . .	$h_2^1(\mathbf{x}) = I\left(\text{wrote} \in \mathbf{x}_1\right)$

Co-boosting NEC Results

- **Data: 90,000 unlabeled named entities**

- **Seeds: Location – New York, California, U.S**
  Person context – Mr.
  Organization name – I.B.M., Microsoft
  Organization context – Incorporated

- **Create labeled data using seeds as rules**
  - **Whenever I see Mr. ____, label it as a person**

- **Results**
  - **Baseline (most frequent) 45% Co-boosting: 91%**

Entity-attribute extraction

Bellare et al. (2008)

- **Entities:** companies, countries, people
- **Attributes:** C.E.O., market capitalization, border, prime minister, age, employer, address
- **Extracting entity-attribute pairs from sentences**

The	**population**	**of**	**China**	**exceeds**
L	**x**	**M**	**y**	**R**

L,M,R = context **x,y** = **content**

Data and learning

- Input: seed list of entities and attributes
  - 2 views: context and content
- Training: co-training decision lists and self-trained MaxEnt classifier
- Problem: No negative instances
  - Just treat all unlabeled instances as negative
  - Re-label most confident positive instances

Examples of learned attributes

- **Countries and attributes**
  - <**Malaysia**, **problems**>, <**Nepal**, **districts**>, <**Colombia**, **important highway**>
- **Companies and attributes**
  - <**Limited Too**, **chief executive**>, <**Intel**, **speediest chip**>, <**Uniroyal**, **former chairman**>

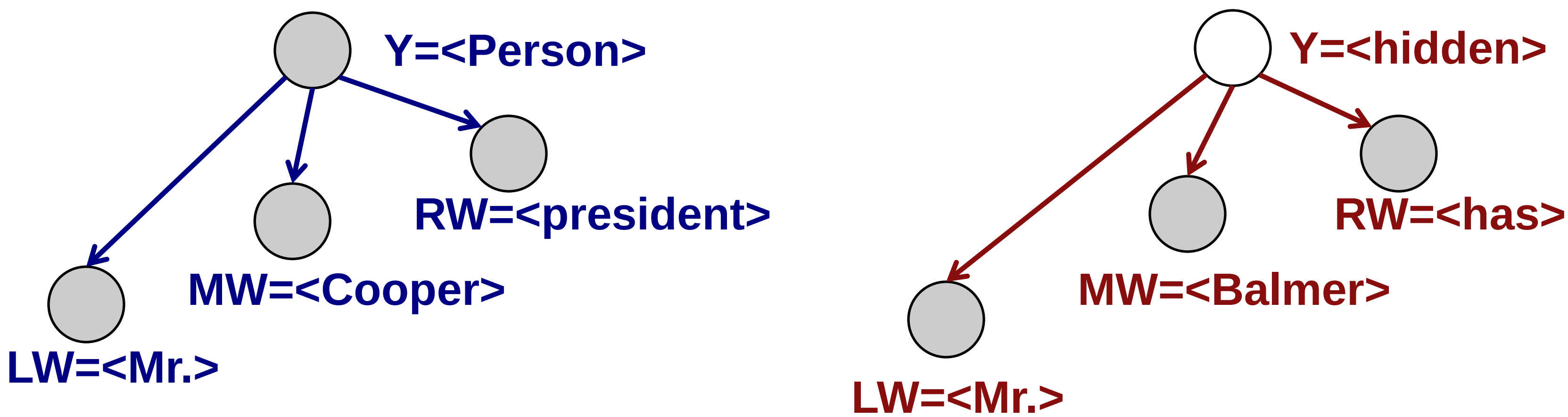
Where can co-boosting go wrong?

- Co-boosting enforces agreement on unlabeled data
- If only 1 view can classify correctly, this causes errors

Co-boosting step	**Data**	**Update action**
Update context view	**Mr.** Balmer has already faxed	$h_1^1(\mathbf{x}) = I\left(\mathrm{Mr.} \in \mathbf{x}_1\right)$
Label unlabeled data	says Mr. **Cooper**, vice president	Label “Person”
Update content view	says Mr. **Cooper**, vice president	$h_1^2(\mathbf{x}) = I\left(\mathrm{Cooper} \in \mathbf{x}_2\right)$
Label unlabeled data	**Cooper Tires** spokesman John	Label “Person”

SSL with latent variables

- **Maximize likelihood treating unlabeled labels as hidden**



- **Labeled data gives us basic label-feature structure. Maximum likelihood (MLE) via EM fills in the gaps**

$$\max_{\theta} \sum_{(\mathbf{x},y;\theta)\in L} \log p(\mathbf{x},y;\theta) + \sum_{\mathbf{x}\in U} \log \left(\sum_{y} p(\mathbf{x},y;\theta) \right)$$

Where can MLE go wrong?

- Unclear when likelihood and error are related

- Collins & Singer (1999) : co-boosting 92%, EM: 83%
- Mark Johnson. Why doesn't EM find good HMM POS-taggers? EMNLP 2007.

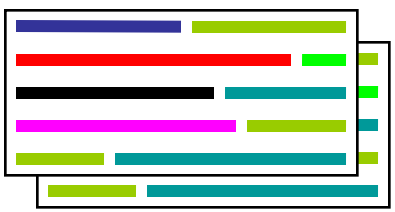
- How can we fix MLE?
  - Good solutions are **high likelihood**, even if they're not **maximum likelihood**
  - **Coming up:** Constrain solutions to be consistent with linguistic intuition

Prototype-driven learning

Haghighi and Klein 2006

Standard SSL

labeled data



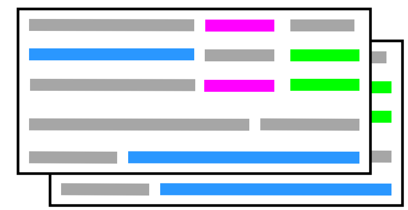
unlabeled data

Prototype learning (part of speech)

prototypes

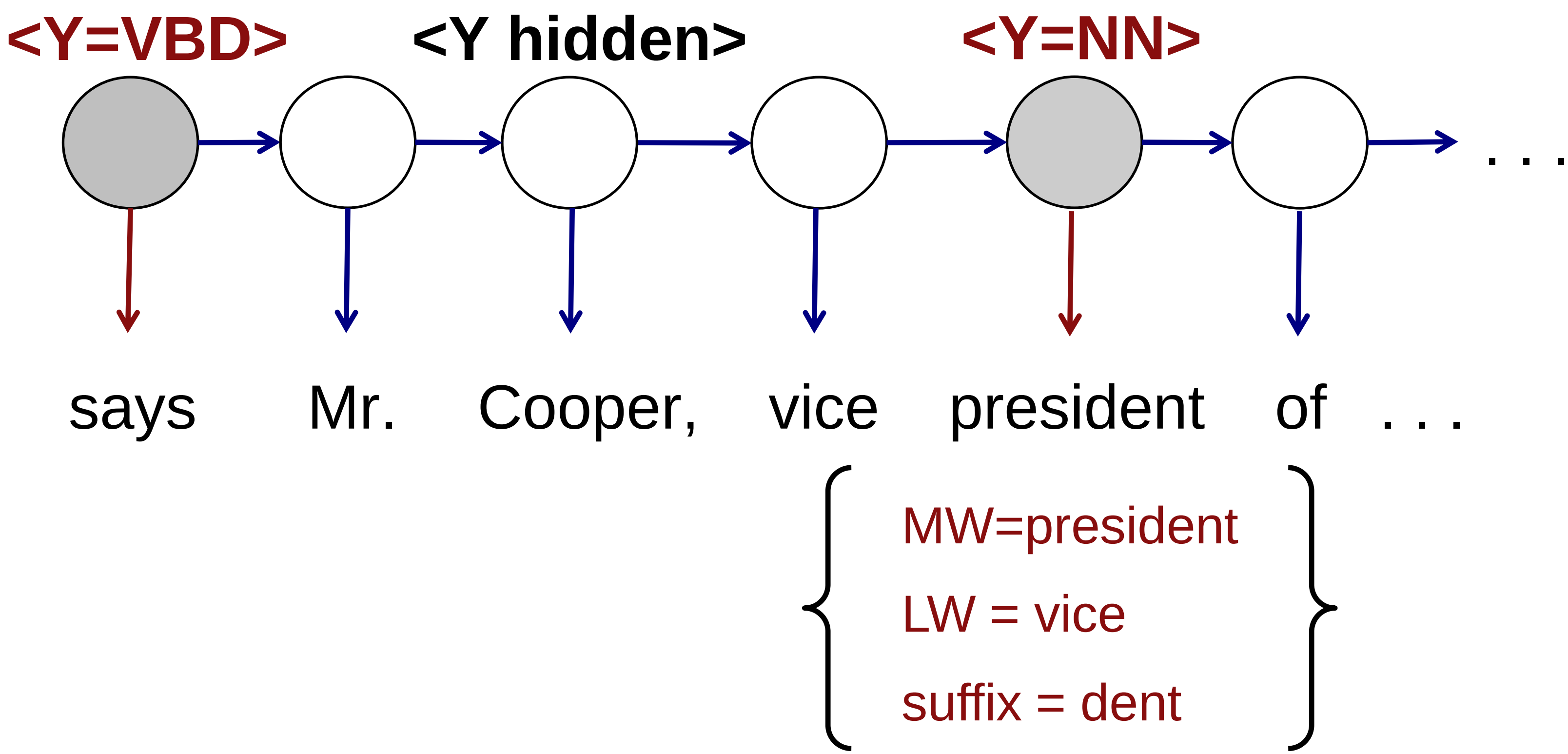
NN	president percent
VBD	said, was had
JJ	new, last, other

training data



- **Each instance is partially labeled**
- **Prototypes force representative instances to be consistent**

Using prototypes in HMMs



- **EM Algorithm:** Constrained forward-backward
- Haghighi and Klein (2006) use Markov random fields

Incorporating distributional similarity

- Represent each word by bigram counts with most frequent words on left & right

president

$$\left(\begin{array}{c} \text{LW="vice": 0.1} \\ \text{LW="the": 0.02} \\ \cdots \\ \text{RW="of": 0.13} \\ \cdots \\ \text{RW="said": 0.05} \end{array}\right)$$

- k-dimensional representation via SVD

$$\left[\begin{array}{c} \mathbf{w}_1 \ldots \mathbf{w}_V \end{array}\right] \approx U_{v \times k} \quad D_{k \times k} \quad V'_{D \times k}$$

- Similarity between a word and prototype

$$\mathrm{sim}(\mathbf{w}_i, \mathbf{pw}_j) = \begin{cases} 1, & \mathbf{w}_i'(UU')\mathbf{pw}_j > \tau \\ 0, & \text{o.t.w.} \end{cases}$$

- We'll see a similar idea when discussing structural learning

Results: Part of speech tagging

Prototype Examples (3 prototypes per tag)

NN	president	**IN**	of	**JJ**	new
VBD	said	**NNS**	shares	**DET**	the
CC	and	**TO**	to	**CD**	million
NNP	Mr.	**PUNC**	.	**VBP**	are

Results

BASE	46.4%
PROTO	67.7%
PROTO+SIM	80.5%

Results: Classified Ads

Goal: Segment housing advertisements

Size Restrict Terms Location

Remodeled 2 Bdrms/1 Bath, **spacious** upper unit, located in Hilltop Mall area. Walking distance to **shopping**, public transportation, and schools. **Paid** water and garbage. No **dogs** allowed.

Prototype examples

LOCATION	near, shopping
TERMS	paid, utilities
SIZE	large, spacious
RESTRICT	dogs, smoking

Results

BASE	46.4%
PROTO	53.7%
PROTO+SIM	71.5%

Computed from bag-of-words in current sentence

Comments on bootstrapping

- Easy to write down and optimize.
- Hard to predict failure cases

- **Co-training** encodes assumptions as 2-view agreement
- **Prototype learning** enforces linguistically consistent constraints on unsupervised solutions

- **Co-training** doesn't always succeed
  - Structural learning section
- **Prototype learning** needs good SIM features to perform well

Entropy and bootstrapping

- **Haffari & Sarkar 2007**. Analysis of Semi-supervised Learning with the Yarowsky Algorithm.
  - Variants of Yarowsky algorithm minimize entropy of p(y | **x**) on unlabeled data.
- Other empirical work has looked at minimizing entropy directly.
- Entropy is not error.
  - Little recent theoretical work connecting entropy & error

More bootstrapping work

- **McClosky & Charniak (2006).** Effective Self-training for Parsing. Self-trained Charniak parser on WSJ & NANC.
- **Aria Haghighi's prototype sequence toolkit.** **http://code.google.com/p/prototype-sequence-toolkit/**
- **Mann & McCallum (2007).** Expectation Regularization. Similar to prototype learning, but develops a regularization framework for conditional random fields.

Graph-based Semi-supervised Learning

- From items to graphs
- Basic graph-based algorithms
  - Mincut
  - Label propagation and harmonic function
  - Manifold regularization
- Advanced graphs
  - Dissimilarities
  - Directed graphs

Text classification: easy example

- Two classes: astronomy vs. travel
- Document = 0-1 bag-of
- Cosine similarity

x1=“bright asteroid”, y

x2=“yellowstone denali”,

x3=“asteroid comet”?

x4=“camp yellowstone”?

Easy, by word overlap

Hard example

x1="bright asteroid", y1=astronomy

x2="yellowstone denali", y2=travel

x3="zodiac"?

x4="airport bike"?

- No word overlap
- Zero cosine similarity
- Pretend you don't know English

Hard example

	x1	x3	x4	x2
asteroid	1			
bright	1			
comet				
zodiac		1		
airport			1	
bike			1	
yellowstone				1
denali				1

Unlabeled data comes to the rescue

	x1	x5	x6	x7	x3	x4	x8	x9	x2
asteroid	1								
bright	1	1	1						
comet		1	1	1					
zodiac				1	1				
airport						1			
bike						1	1	1	
yellowstone							1	1	1
denali								1	1

Intuition

1. Some **unlabeled documents** are similar to the **labeled documents** → same label

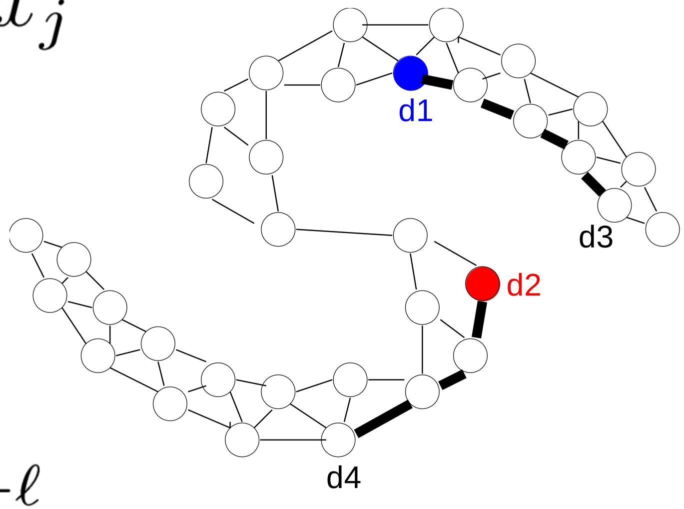
2. Some **other unlabeled documents** are similar to the above **unlabeled documents** → same label

3. ad infinitum

We will formalize this with graphs.

The graph

- Nodes $\{x_1, \ldots, x_\ell\} \cup \{x_{\ell+1}, \ldots, x_{m+\ell}\}$
- Weighted, undirected edges w_{ij}
  - Large weight ➔ similar x_i, x_j
- Known labels $y_1, \ldots, y_\ell$
- Want to know
  - transduction: $y_{\ell+1}, \ldots, y_{m+\ell}$
  - induction: y^* for new test item x^*



How to create a graph

- **Empirically, the following works well:**

1. **Compute distance between *i*, *j***
2. **For each *i*, connect to its kNN. k very small but still connects the graph**
3. **Optionally put weights on (only) those edges**

$$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

4. **Tune σ**

Mincut (*st*-cut)

Binary labels $y_i \in \{0, 1\}$. Fix $Y_l = \{y_1, \ldots, y_\ell\}$.

Solve for $Y_u = \{y_{\ell+1}, \ldots, y_{\ell+m}\}$:

$$\min_{Y_u} \sum_{i,j=1}^{n} w_{ij}(y_i - y_j)^2$$

Combinatorial problem (integer program), but efficient polynomial time solver (Boykov,Veksler,Zabih PAMI 2001).

Mincut example: subjectivity

- **Task:** classify each sentence in a document into **objective**/**subjective**. (Pang,Lee. ACL 2004)
- NB/SVM for isolated classification
  - Subjective data (y=1): Movie review snippets "bold, imaginative, and impossible to resist"
  - Objective data (y=0): IMDB
- But there is more…

Mincut example: subjectivity

- Key observation: sentences next to each other tend to have the same label

  $w_{ij} = c$ if x_i, x_j are close, 0 otherwise.

- Two special labeled nodes (source, sink)

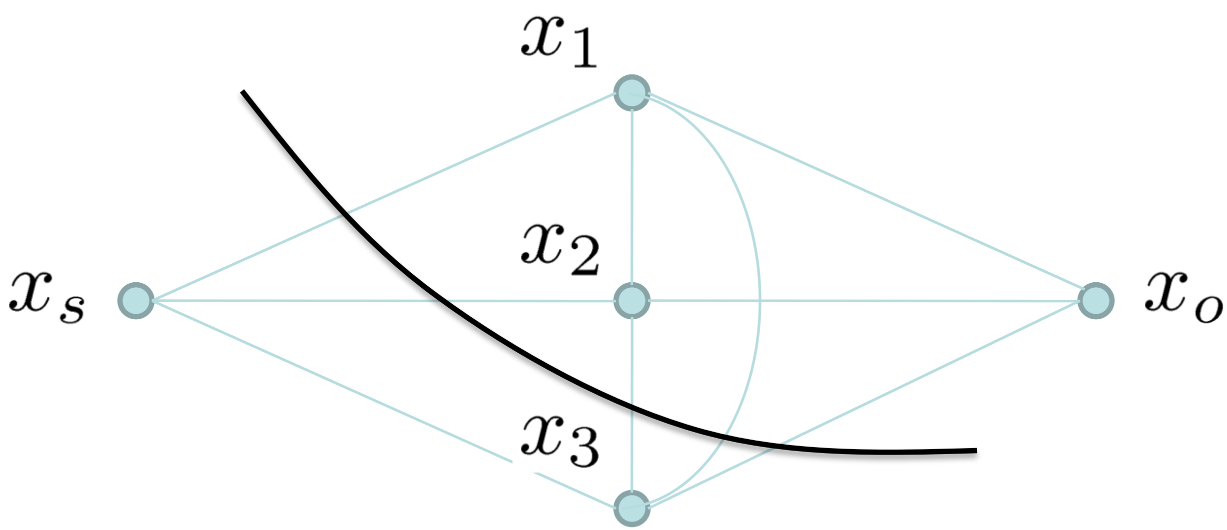
  $(x_s, y_s = 1), (x_o, y_o = 0)$

- Every sentence connects to both:

  $$w_{si} = Pr(y_i = 1 | x_i, NB)$$
  $$w_{io} = Pr(y_i = 0 | x_i, NB)$$

Mincut example: subjectivity

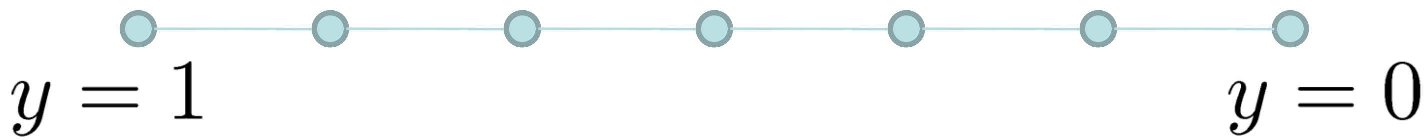
$\min \sum_{ij} w_{ij}(y_i - y_j)^2$ minimizes the cut



$$\sum_{ij:y_i \neq y_j} w_{ij}$$

Some issues with mincut

- Multiple equally min cuts, but different



- Lacks classification confidence
- These are addressed by harmonic functions and label propagation

Harmonic Function

Relax $\{0, 1\}$ labels to real values $f(x) \in \mathbb{R}$.

$f(x_\ell) = y_\ell$.

$$\min_{f_u} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

Same as mincut except that $f_u \in \mathbb{R}$.

The harmonic function is the solution f_u.
Unique. $f_u \in [0, 1]$ less confident near 0.5.

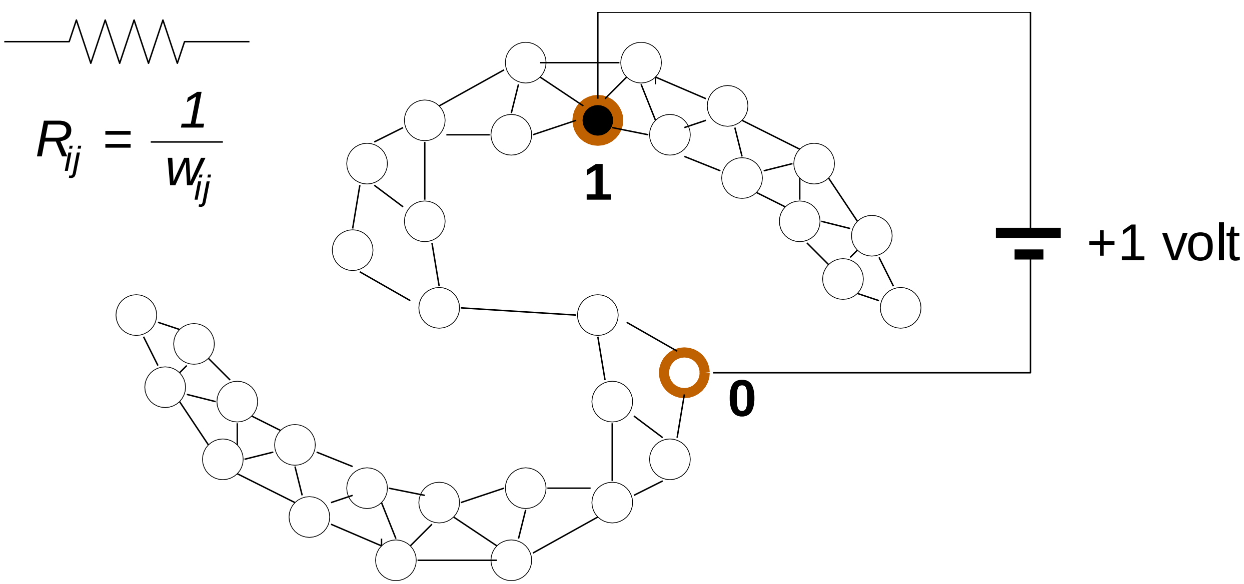
An electric network interpretation

Edges has conductance w_{ij}

1-volt battery connects to labeled points y_ℓ

Voltage at node $i = f_i$

Similar voltage if many strong paths exist.



Label propagation

Naïve algorithm for the harmonic function:

1. Fix $f_\ell = y_\ell$. Set $f_u = 0$ (arbitrary)
2. Repeat: $f_u = \frac{\sum_{i \sim u} w_{iu} f_i}{\sum_{i \sim u} w_{iu}}$

Converges but slow. Better optimize directly.

The graph Laplacian

- W: $n \times n$ weight matrix.
- D: degree matrix $d_{ii} = \sum_{j=1}^{n} w_{ij}$, diagonal
- Unnormalized graph Laplacian $L = D - W$
- Energy $\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 = f^\top L f$

$$\min_{f_u} f^\top L f$$
$$\text{s.t.} f_\ell = y_\ell$$

Closed-form solution

Partition the Laplacian $L = \left[\begin{array}{cc} L_{\ell\ell} & L_{\ell u} \\ L_{u\ell} & L_{uu} \end{array} \right].$

Harmonic function (=label propagation)

$$f_u = -L_{uu}^{-1} L_{ul} y_\ell.$$

Can use the normalized Laplacian too:

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

Harmonic example 1: WSD

- WSD from context, e.g., “interest”, “line” (Niu,Ji,Tan ACL 2005)
- x_i: context of the ambiguous word, features: POS, words, collocations
- d_{ij}: cosine similarity or JS-divergence
- w_{ij}: kNN graph
- Labeled data: a few x_i’s are tagged with their word sense.

Harmonic example 1: WSD

- SENSEVAL-3, as percent labeled:

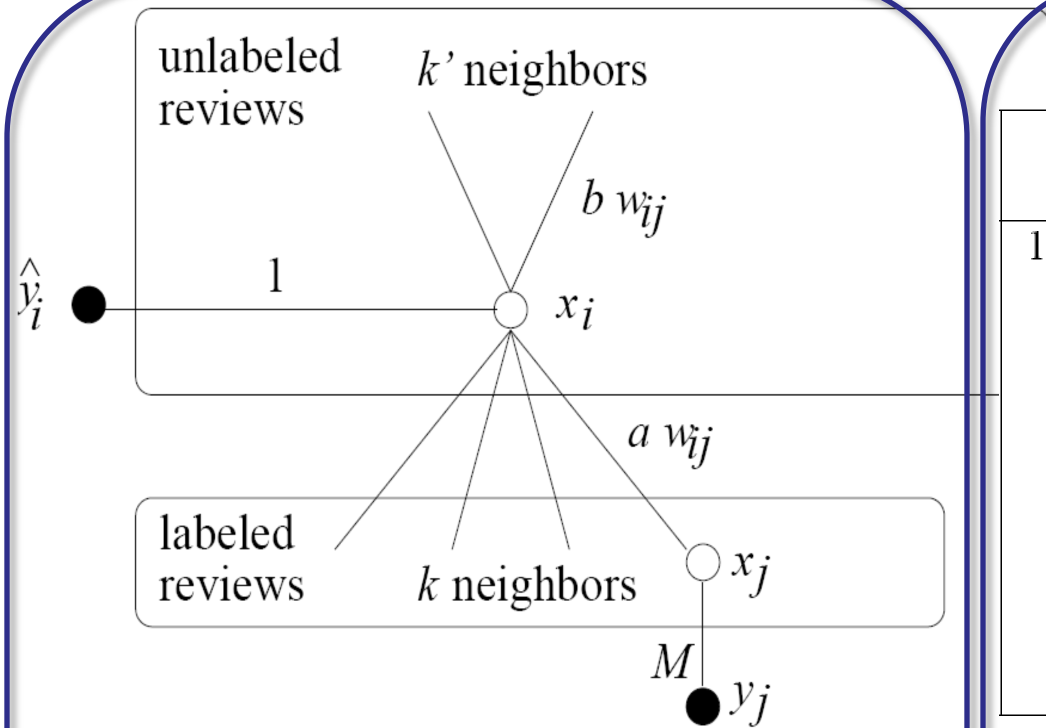
Percentage	SVM	LP_{cosine}	LP_{JS}
1%	24.9±2.7%	27.5±1.1%	28.1±1.1%
10%	53.4±1.1%	54.4±1.2%	54.9±1.1%
25%	62.3±0.7%	62.3±0.7%	63.3±0.9%
50%	66.6±0.5%	65.7±0.5%	66.9±0.6%
75%	68.7±0.4%	67.3±0.4%	68.7±0.3%
100%	69.7%	68.4%	70.3%

(Niu,Ji,Tan ACL 2005)

Harmonic example 2: sentiment

- Rating (0-3) from movie reviews (Goldberg,Zhu. NAACL06 workshop)
- x_i: movie reviews
- w_{ij}: cosine similarity btw "positive sentence percentage" (PSP) vectors of $x_{i,}$ x_j
- PSP classifier trained on "snippet" data (Pang,Lee. ACL 2005)

Harmonic example 2: sentiment



Graph

$\lvert L \rvert$	regression	PSP reg+PSP	PSP SSL+PSP
1593	**0.592**	**0.592**	0.546
800	**0.553**	**0.554**	0.534
400	**0.522**	**0.525**	**0.526**
200	0.494	0.498	**0.521**
100	0.463	0.477	**0.511**
50	0.439	0.458	**0.499**
25	0.408	0.421	**0.465**
12	0.401	0.378	**0.451**
6	0.390	0.359	**0.422**

Accuracy

Some issues with harmonic function

- It fixes the given labels y_l
  - What if some labels are wrong?

- It cannot easily handle new test items
  - Transductive, not inductive
  - Add test items to graph, recompute

- Manifold regularization addresses these issues

Manifold regularization

SVM: $\min_f \sum_{i=1}^{\ell} \max(1 - y_i f_i, 0) + \lambda \|f\|^2$

$f \in \mathrm{RKHS}(K)$ defined everywhere.

SVM with manifold regularization:

$$\min_f \sum_{i=1}^{\ell} \max(1 - y_i f_i, 0) + \lambda_1 \|f\|^2 + \lambda_2 f_{1:\ell+m}^\top L f_{1:\ell+m}$$

Label noise OK (slack).

Classify new test item x by $\mathrm{sgn}(f(x))$.

Manifold example

- Text classification (Sindhwani,Niyogi,Belkin.ICML 2005)
- x_i: mac/windows. TFIDF.
- w_{ij}: weighted kNN graph $\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

Dataset →	mac-win
Algorithm ↓	unlab test
SVM	20.9 20.9
LapSVM	9.9 9.7

$$l = 50, u = 1411, \text{ test=485}$$

Advanced topics

- So far edges denote symmetric similarity
  - Larger weights ➔ similar labels

- What if we have dissimilarity knowledge?
  - "Two items probably have different labels"

- What if the relation is asymmetric?
  - x_i related to x_j but x_j not always related to x_i

Dissimilarity

- Political view classification
  (Goldberg, Zhu, Wright. AISTATS 2007)

> deshrubinator: “You were the one who thought it should be investigated last week.”

Dixie: No I didn’t, and I made it clear. You are insane! YOU are the one with NO ****ING RESPECT FOR DEMOCRACY!

- They disagree ➔ different classes
- Indicators: quoting, !?, all caps (internet shouting), etc.

Dissimilarity

- Recall to encode similarity between i,j:

$$\min w_{ij}(f_i - f_j)^2$$

- Wrong ways: small w = no preference; negative w nasty optimization

- One solution (also see (Tong,Jin.AAAI07))

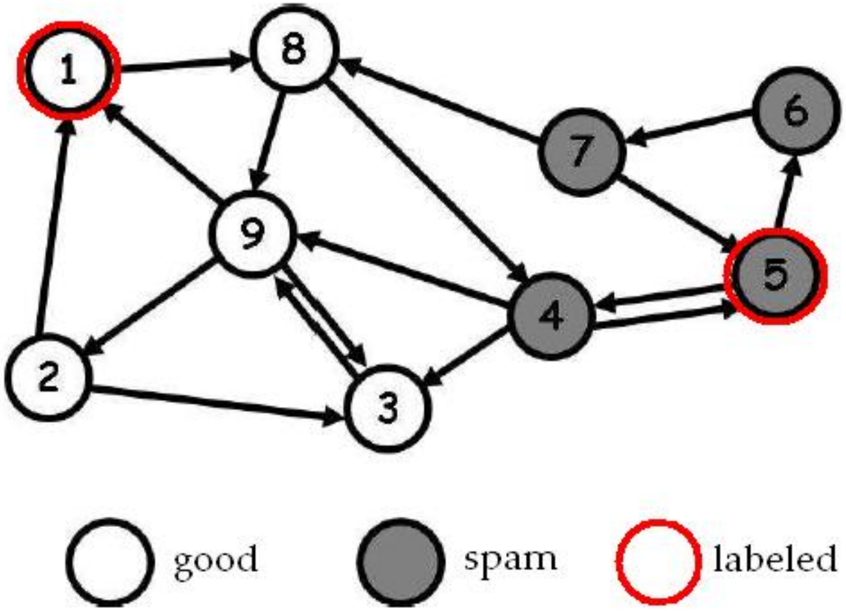
$$\min w_{ij}(f_i + f_j)^2, \text{ note } y \in \{-1, 1\}$$

- Overall $\min \sum_{ij} w_{ij}(f_i \pm f_j)^2$

depends on dissim, sim

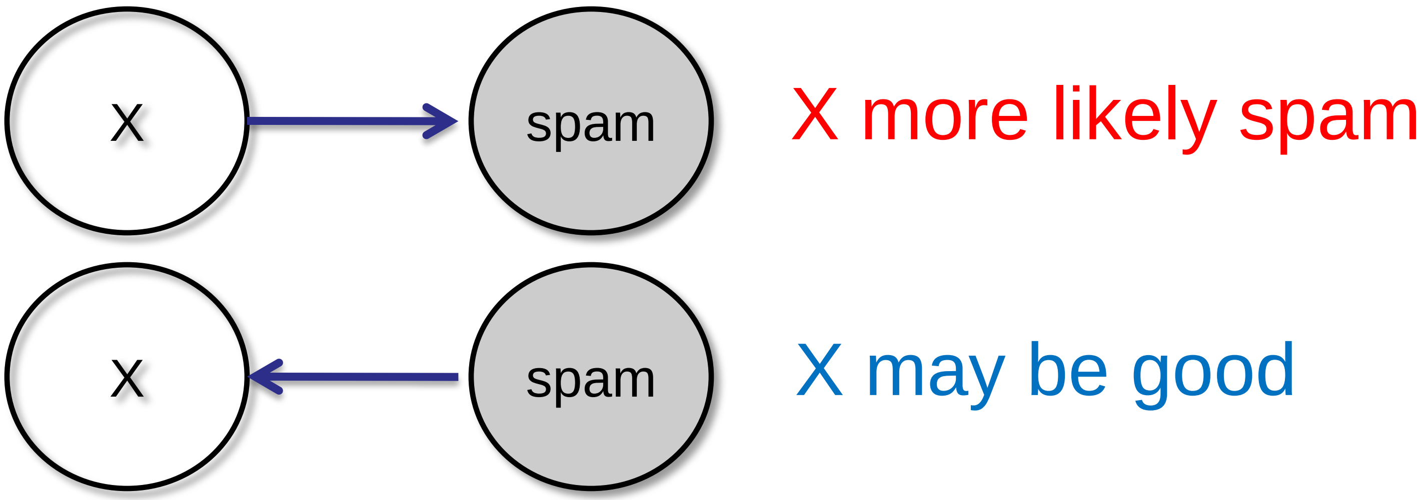
Directed graphs

- Spam vs. good webpage classification (Zhou,Burges,Tao. AIRW 2007)
- Hyperlinks as graph edges, a few webpages manually labeled



Directed graphs

- Directed hyperlink edges



- Can define an analogous “directed graph Laplacian” + manifold regularization

Caution

- **Advantages of graph-based methods:**
  - **Clear intuition, elegant math**
  - **Performs well if the graph fits the task**
- **Disadvantages:**
  - **Performs poorly if the graph is bad: sensitive to graph structure and edge weights**
  - **Usually we do not know which will happen!**

Structural learning: outline

- The structural learning algorithm
- Application to named entity recognition
- Domain adaptation with structural correspondence learning
- Relationship between structural and two-view learning

Structural learning

- **Ando and Zhang (2005)**. Use unlabeled data to constrain structure of hypothesis space

- Given a **target problem** (entity classification)
- Design **auxiliary problems**
  - Look like target problem
  - Can be trained using unlabeled data

- Regularize target problem hypothesis to be close to auxiliary problem hypothesis space

What are auxiliary problems?

2 criteria for auxiliary problems

1) Look like target problem
2) Can be trained from unlabeled data

Named entity classification: Predict presence or absence of left / middle / right words

Left	Middle	Right
Mr.	Thursday	Corp.
President	John	Inc.
	York	said

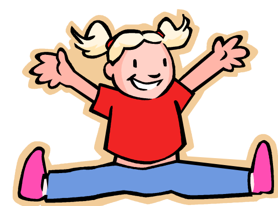
Auxiliary problems for sentiment classification

Running with Scissors: A Memoir

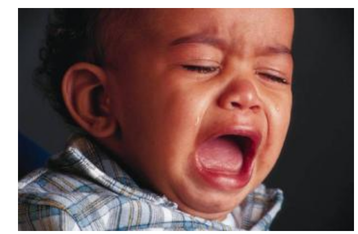
Title: Horrible book, horrible.

This book was horrible. I read half of it, **suffering** from a headache the entire time, and eventually i lit it on fire. One less copy in the world... **don't waste** your money. I wish i had the time spent reading this book back so i could use it for better purposes. This book wasted my life

Labels



Positive



Negative

Auxiliary Problems

Presence or absence of frequent words and bigrams

don't_waste, horrible, suffering

Auxiliary problem hypothesis space

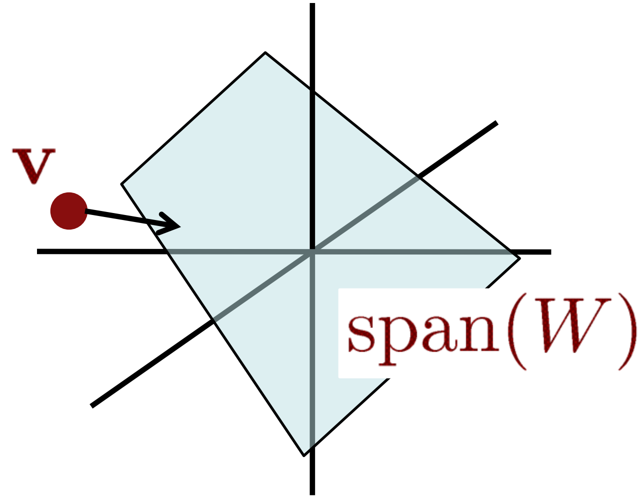
Consider linear, binary auxiliary predictors:

$f_i(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}_i' \, \mathbf{x})$

$$\begin{bmatrix} | & & | & & \\ \mathbf{w}_1 & \cdots & \mathbf{w}_i & \cdots & \mathbf{w}_n \\ | & & | & & \end{bmatrix}$$

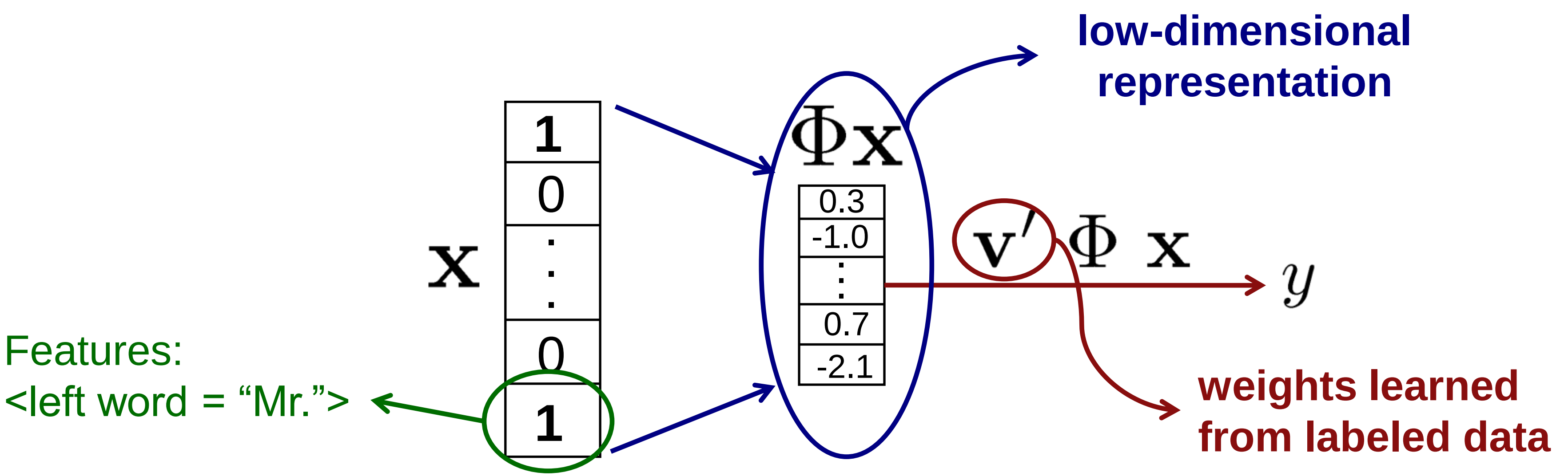
weight vector for auxiliary problem i

Given a new hypothesis weight vector $\mathbf{v}$, how far is it from $\mathrm{span}(W)$?



Two steps of structural learning

Step 1: Use unlabeled data and auxiliary problems to learn a representation $\mathbf{\Phi}$: an approximation to $\mathrm{span}(W)$



Step 2: Use labeled data to learn weights for the new representation

Unlabeled step: train auxiliary predictors

For each unlabeled instance, create a binary presence / absence label

(1) The book is so **repetitive** that I found myself yelling …. I will definitely [redacted] another.

(2) An **excellent** book. Once again, another wonderful novel from Grisham

Binary problem: Does "**not buy**" appear here?

- **Mask** and predict pivot features using other features
- Train n **linear predictors**, one for each binary problem
- Auxiliary weight vectors give us clues about feature conditional **covariance structure**

Unlabeled step: dimensionality reduction

$$\begin{bmatrix} | & & | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_i & \cdots & \mathbf{w}_n \\ | & & | & & | \end{bmatrix}$$

- $\mathbf{W}'\mathbf{x}$ gives n new features

- value of i^{th} feature is the propensity to see **"not buy"** in the same document

- **We want a low-dimensional representation**

- **Many pivot predictors give similar information**
  - **"horrible", "terrible", "awful"**

- **Compute SVD & use top left singular vectors Φ**

Step 2: Labeled training

<u>Step 2</u>: Use Φ to regularize labeled objective

$$\min_{\mathbf{v},\mathbf{w}} \sum_{\mathbf{x},y} L\left(\left(\mathbf{w}'\mathbf{x} + \mathbf{v}'\Phi\mathbf{x}, y\right) + \lambda||\mathbf{w}||_2^2\right.$$

Original, high-dimensional weight vector

low-dimensional weight vector for learned features

Only high-dimensional features have quadratic regularization term

Step 2: Labeled training

$$\sum_{\mathbf{x},y} L\left(\left(\mathbf{w}'\mathbf{x} + \mathbf{v}'\Phi\mathbf{x}, y\right)\right) + \lambda||\mathbf{w}||_2^2$$

- **Comparison to prototype similarity**

  — Uses predictor (weight vector) space, rather than counts

  — Similarity is learned rather than fixed

Results: Named entity recognition

- **Data: CoNLL 2003 shared task**
  - Labeled: 204 thousand tokens of Reuters news data
  - Annotations: person, location, organization, miscellaneous
  - Unlabeled: 30 million words of Reuters news data

- **A glance of some of the rows of Φ**

ROW #	Features
4	Ltd, Inc, Plc, International, Association, Group
9	PCT, N/A, Nil, Dec, BLN, Avg, Year-on-Year
11	San, New, France, European, Japan
15	Peter, Sir, Charles, Jose, Paul, Lee

Numerical Results (F-measure)

Data size / Model	10k tokens	204k tokens
Baseline	72.8	85.4
Co-training	73.1	85.4
Structural	81.3	89.3

- **Large difference between co-training here and co-boosting (Collins & Singer 1999)**
- **This task is entity recognition, not classification**
- **We must improve over a supervised baseline**

Domain adaptation with structural learning

Blitzer et al. (2006): Structural Correspondence Learning (SCL)

Blitzer et al. (2007): For sentiment: **books** & **kitchen appliances**

Running with Scissors: A Memoir

Title: Horrible book, horrible.

This book was horrible. I **read half** of it, **suffering from a headache** the entire time, ...

less copy in the world... don't waste your money. I wish i had the time spent reading this book back so i could use it for better purposes. This book wasted my life

Avante Deep Fryer, Chrome & Black

Title: lid does not work well...

I love the way the Tefal deep fryer cooks, however, I am **returning** my ...

closure. The lid may close initially, but after a few uses it no longer stays closed. I will not be purchasing this one again.

Error increase: 13% → 26%

Pivot Features

Pivot features are features which are shared across domains

Unlabeled **kitchen** contexts	Unlabeled **books** contexts
• Do **not buy** the Shark portable steamer …. Trigger mechanism is **defective**.	• The book is so **repetitive** that I found myself yelling …. I will definitely **not buy** another.
• the very nice lady assured me that I must have a **defective** set …. What a **disappointment**!	• A **disappointment** …. Ender was talked about for **<#> pages** altogether.
• Maybe mine was **defective** …. The directions were **unclear**	• it's **unclear** …. It's repetitive and **boring**

Use presence of pivot features as auxiliary problems

Choosing pivot features: mutual information

Pivot selection (SCL): Select top features x_i by shared counts

Pivot selection (SCL-MI): Select top features in two passes

(1) Filter feature x_i if min count in both domains < k

(2) Select top filtered features by $\mathrm{PMI}(x_i, y)$

Books-kitchen example

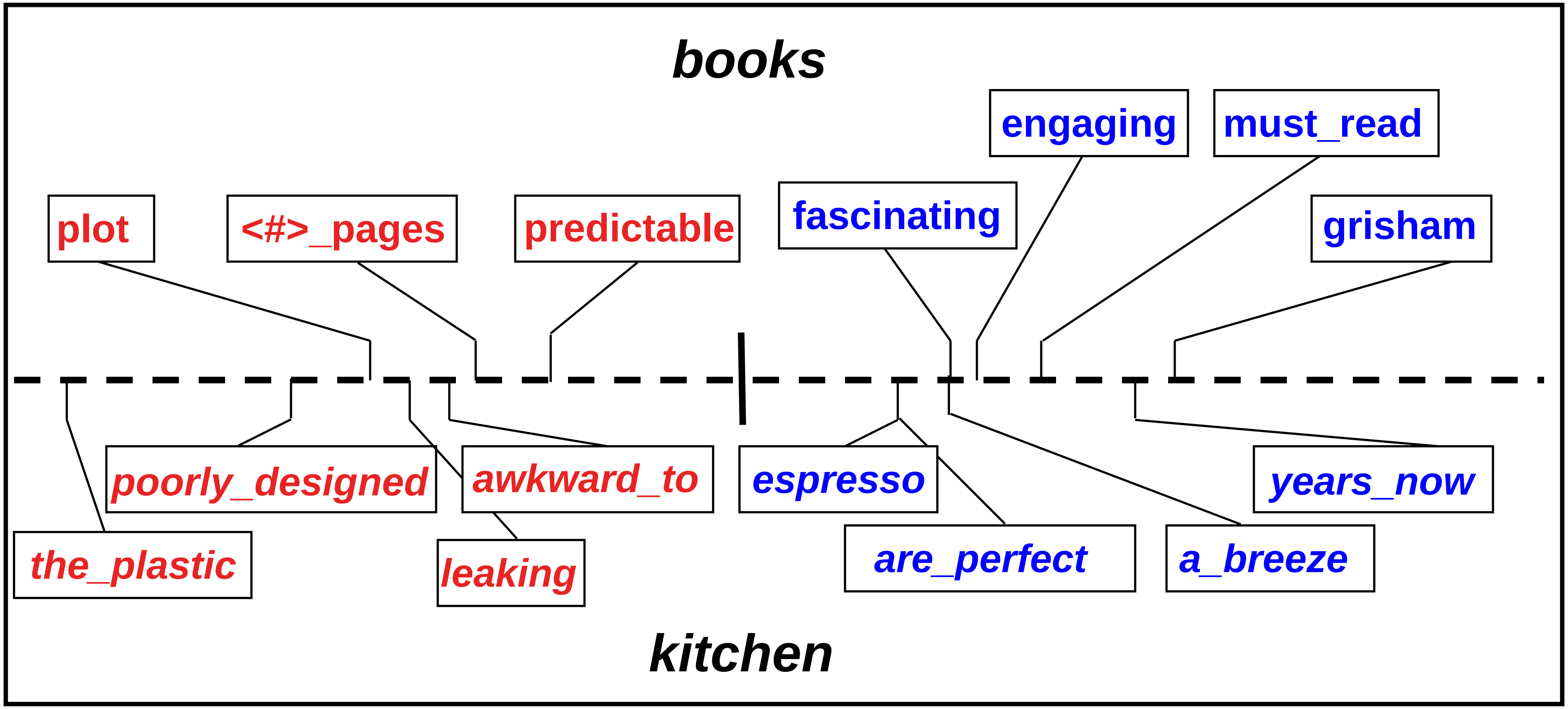
In SCL, not SCL-MI	In SCL-MI, not SCL
book one <num> so all very about they like good when	**a_must a_wonderful loved_it weak don't_waste awful highly_recommended and_easy**

Sentiment Classification Data

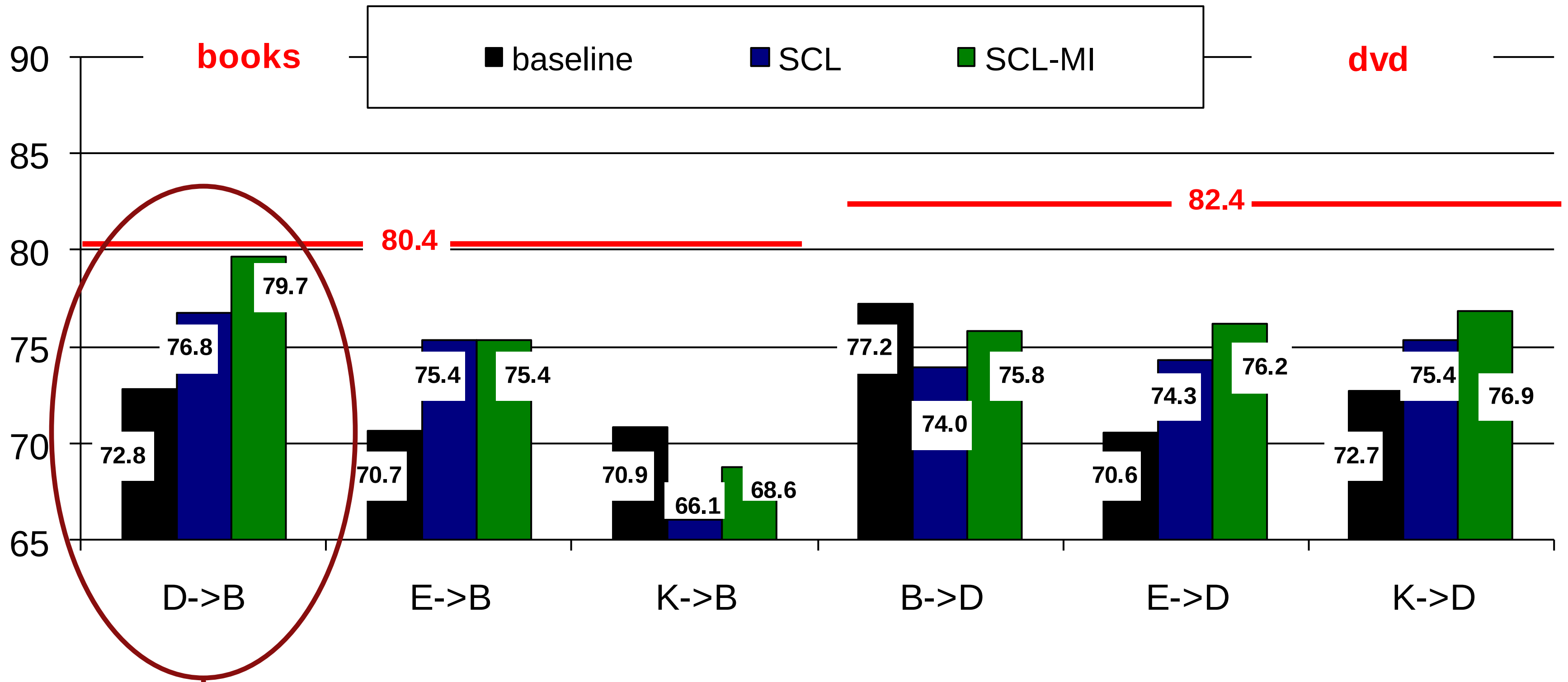
- **Product reviews from Amazon.com**
  - Books, DVDs, Kitchen Appliances, Electronics
  - 2000 labeled reviews from each domain
  - 3000 – 6000 unlabeled reviews
- **Binary classification problem**
  - Positive if 4 stars or more, negative if 2 or less
- **Features:** unigrams & bigrams
- **Pivots:** SCL & SCL-MI
- **At train time:** minimize Huberized hinge loss (Zhang, 2004)

Visualizing Φ (books & kitchen)

negative **vs.** **positive**



Empirical Results: books & DVDs



on average, scl-mi reduces error due to adaptation by 36%

baseline loss due to adaptation: 7.6%

SCL-MI loss due to adaptation: 0.7%

Structural learning: Why does it work?

- Good auxiliary problems = good representation

- Structural learning vs. co-training
  - **Structural learning separates unsupervised and supervised learning**
  - **Leads to a more stable solution**

- Structural learning vs. graph regularization
  - **Use structural learning when auxiliary problems are obvious, but graph is not**

Understanding structural learning: goals

- Develop a relationship between structural learning and multi-view learning
- Discuss assumptions under which structural learning can perform well
- Give a bound on the error of structural learning under these assumptions

Structural and Multi-view learning

Context pivots	**Orthography features**	**Context features**	**Orthography pivots**
$\mathbf{X}^{(1)}$	$\mathbf{X}^{(2)}$	$\mathbf{X}^{(1)}$	$\mathbf{X}^{(2)}$

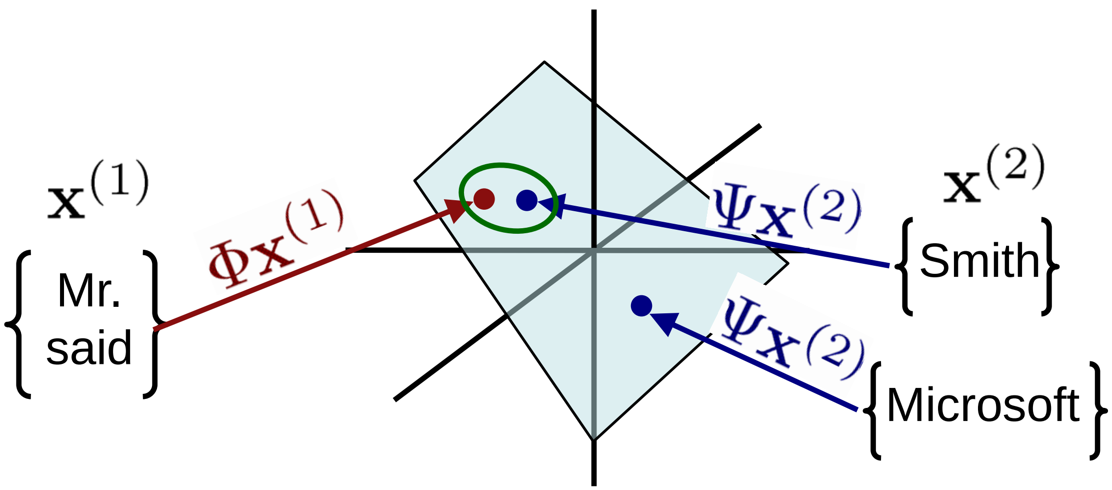
1. Learn W, the matrix of pivot predictors
2. Let Φ be the top k left singular vectors of W

1. Learn V, the matrix of pivot predictors
2. Let Ψ be the top k left singular vectors of V

Canonical correlation analysis

Canonical correlation analysis – CCA (Hotelling, 1936)

- Dimensionality reduction for jointly distributed random variables $(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \sim \mathcal{D}$
- CCA yields matrices $\Psi, \Phi \in \mathbb{R}^{d \times k}$ such that $\Psi\mathbf{X}^{(1)}$ and $\Phi\mathbf{X}^{(2)}$ are maximally correlated



Correlated features from different views are mapped to similar areas of space

Structural learning and CCA

Some changes to structural learning

(1) Minimize squared loss for auxiliary predictors

(2) Block SVD by view: Train auxiliary predictors for view 1 using features from view 2 and vice versa

Let W_1, W_2 be the matrices of modified auxiliary predictors for views 1 and 2

If the matrices Φ and Ψ are the top left singular vectors of W_1, W_2, then these are exactly the Φ and Ψ from CCA

CCA and semi-supervised learning

Kakade and Foster (2007). Multi-view regression via canonical correlation analysis.

Contrast with co-training: K&F **don't** assume independence

Assume:

The best model $\mathbf{w}^{(\nu)}$ for each view has low regret to the best joint linear model $\mathbf{w}$.

$$E\left[(\mathbf{w}^{(\nu)}\mathbf{x} - y)^2 - (\mathbf{w}\mathbf{x} - y)^2\right] \leq \epsilon$$

Semi-supervised learning procedure

On unlabeled data, compute CCA. Let Φ be the CCA transformation for view 1.

CCA also yields correlation coefficients $\lambda_i \in [0,1]$ with $\lambda_{i+1} \leq \lambda_i$

Sum of correlation coefficients indicates total amount of correlation

Regularize based on amount of correlation

Training error using transformed inputs

Let $\hat{\mathbf{v}}^{(1)} = \arg\min_{\mathbf{v}^{(1)}} \sum_{i=1}^{\ell} (\hat{\mathbf{v}}^{(1)} \Phi \mathbf{x}_i - y_i)^2 + \sum_j \frac{1-\lambda_j}{\lambda_j} v_j^2$

A bound on squared error under CCA

Main theorem of Kakade & Foster (2007)

$$\text{Let } \lambda_j \text{ be the } j^{\text{th}} \text{ correlation coefficient. Then}$$

$$E(\hat{\mathbf{v}}^{(1)}\Phi\mathbf{x} - y)^2 - E(\mathbf{wx} - y)^2 \leq 5\epsilon + \frac{1}{\ell}\sum_j \lambda_j^2$$

Expected error of learned, transformed predictor

Expected error of best model

Assumption: How good is single view compared to joint model?

number of training examples

amount of correlation

When can structural learning break?

- Hard-to-define auxiliary problems
  - Dependency parsing: How to define auxiliary problems for an edge?
  - MT alignment: How to define auxiliary problems for a pair of words?

- Combining real-valued & binary features

high-dimensional, sparse

$$\left[\,-\;-\;-\;\mathbf{x}\;-\;-\;-\;\middle|\;-\;\Phi\mathbf{x}\;-\,\right]$$

low-dimensional, dense

  - scaling, optimization

Other work on structural learning

- Scott Miller et al. (2004). Name Tagging with Word Clusters and Discriminative Training.
  - Hierarchical clustering, not structural learning.
  - Representation easily combines with binary features

- Rie Ando, Mark Dredze, and Tong Zhang (2005). TREC 2005 Genomics Track Experiments at IBM Watson.
  - Applying structural learning to information retrieval

- Ariadna Quattoni, Michael Collins, and Trevor Darrel (CVPR 2007). Learning Visual Representations using Images with Captions.

SSL Summary

- **Bootstrapping**
  - Easy to write down. Hard to analyze.
- **Graph-based Regularization**
  - Works best when graph encodes information not easily represented in normal feature vectors
- **Structural Learning**
  - With good auxiliary problems, can improve even with lots of training data
  - Difficult to combine with standard feature vectors

Two take-away messages

1) Semi-supervised learning yields good results for small amounts of labeled data

2) “I have lots of labeled data” is not an excuse not to use semi-supervised techniques

http://ssl-acl08.wikidot.com