CS 4650/7650
Semi-Supervised Learning[1]

Jacob Eisenstein

November 13, 2014

[1] With slides borrowed from John Blitzer, Mingwei Chang, Hal Daume III, Xiaojin Zhu, and everyone they borrowed slides from...

Frameworks for learning

- So far, we have focused on learning a function f from labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$.
- What if you don't have labeled data for a domain or task you want to solve?
  - You can use labeled data from another domain.
    This rarely works well.
  - You can label data yourself.
    This is a lot of work.

Examples

Phonetic transcription[2]

- "Switchboard" dataset of telephone conversations
- Annotations from word to phoneme sequence:
  - film $\rightarrow$ F IH_N UH_GL_N M
  - be all $\rightarrow$ BCL B IY IY_TR AO_TR AO L_DL

[2] Examples from Xiaojin "Jerry" Zhu

Examples

Phonetic transcription[2]

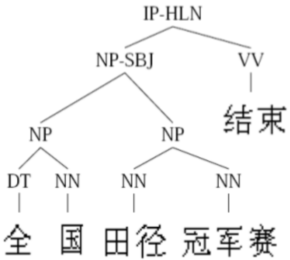
- "Switchboard" dataset of telephone conversations
- Annotations from word to phoneme sequence:
  - film → F IH_N UH_GL_N M
  - be all → BCL B IY IY_TR AO_TR AO L_DL
- **400 hours** annotation time per hour of speech!

[2] Examples from Xiaojin "Jerry" Zhu

Examples

Natural language parsing[3]

- Penn Chinese Treebank
- Annotations from word sequences to parse trees



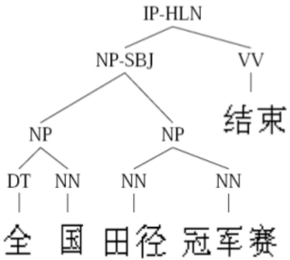
"The National Track and Field Championship has finished."

[3]Examples from Xiaojin "Jerry" Zhu

Examples

Natural language parsing[3]

- Penn Chinese Treebank
- Annotations from word sequences to parse trees



"The National Track and Field Championship has finished."

- **2 years** annotation time for 4000 sentences

[3]Examples from Xiaojin "Jerry" Zhu

How can we learn with less annotation effort?

How can we learn with less annotation effort?

- **Semisupervised learning**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$: labeled examples
  - $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$: unlabeled examples
  - often $u \gg \ell$

How can we learn with less annotation effort?

- **Semisupervised learning**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$: labeled examples
  - $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$: unlabeled examples
  - often $u \gg \ell$

- **Domain adaptation**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$: labeled examples in *source* domain
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_T} \sim \mathcal{D}_T$: labeled examples in *target* domain
  - possibly some unlabeled data in target and possibly source domain
  - evaluate in the target domain

How can we learn with less annotation effort?

- **Semisupervised learning**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$: labeled examples
  - $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$: unlabeled examples
  - often $u \gg \ell$
- **Domain adaptation**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$: labeled examples in *source* domain
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_T} \sim \mathcal{D}_T$: labeled examples in *target* domain
  - possibly some unlabeled data in target and possibly source domain
  - evaluate in the target domain
- **Active learning**: model can query annotator for labels

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'intrigues

How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'intrigues

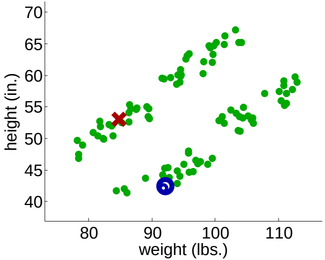
How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

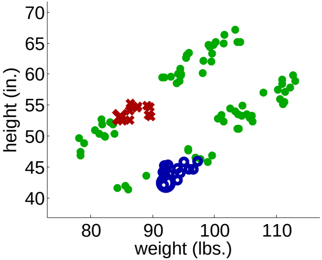
- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'intrigues

By propagating training labels to unlabeled data, we learn the sentiment value of many more words.

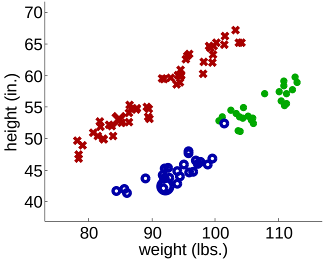
Propagating 1-Nearest-Neighbor: now it works



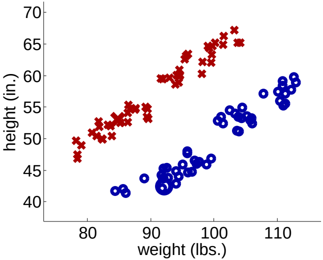
(a) Iteration 1



(b) Iteration 25



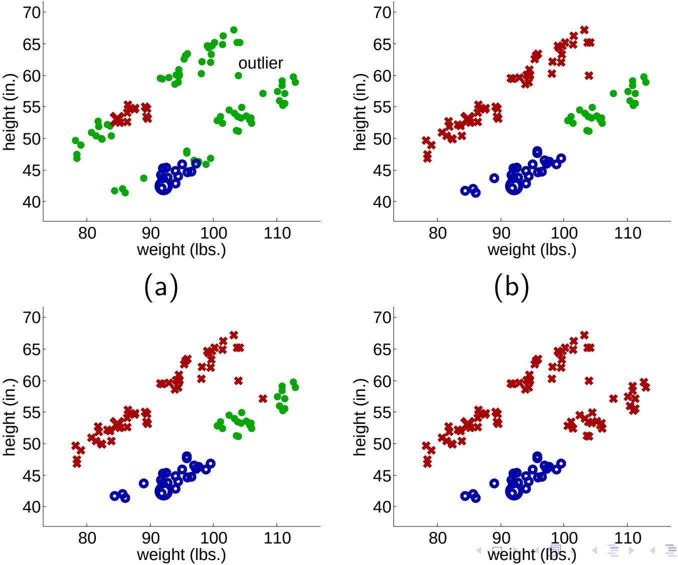
(c) Iteration 74



(d) Final labeling of all instances

Propagating 1-Nearest-Neighbor: now it doesn't

But with a single outlier...



When does bootstrapping work? “Folk wisdom”

- Better for generative models (e.g., Naive Bayes) than for discriminative models (e.g., perceptron)
- Better when the Naive Bayes assumption is stronger.
  - Suppose we want to classify NEs as PERSON or LOCATION
  - Features: string and context, e.g.
    - located on Peachtree Street
    - Dr. Walker said ...

When does bootstrapping work? "Folk wisdom"

- Better for generative models (e.g., Naive Bayes) than for discriminative models (e.g., perceptron)
- Better when the Naive Bayes assumption is stronger.
  - Suppose we want to classify NEs as PERSON or LOCATION
  - Features: string and context, e.g.
    - located on Peachtree Street
    - Dr. Walker said ...

$$P(x_1 = \text{street}, x_2 = \text{on} | \textsc{loc}) \approx P(x_1 = \text{street} | \textsc{loc}) P(x_2 = \text{on} | \textsc{loc})$$

Two views and co-training

- **Co-training** makes bootstrapping folk wisdom explicit.
  - Assume two, **conditionally independent**, views of a problem.
  - Assume each view is sufficient to do good classification.

Two views and co-training

- **Co-training** makes bootstrapping folk wisdom explicit.
  - Assume two, **conditionally independent**, views of a problem.
  - Assume each view is sufficient to do good classification.

- Sketch of learning algorithm:
  - On labeled data, minimize error.
  - On unlabeled data, constrain the models from different views to agree with each other.

Co-training example

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1.	Peachtree Street	located on	LOC
2.	Dr. Walker	said	PER
3.	Zanzibar	located in	?
4.	Zanzibar	flew to	?
5.	Dr. Robert	recommended	?
6.	Oprah	recommended	?

Algorithm

Co-training example

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1.	Peachtree Street	located on	LOC
2.	Dr. Walker	said	PER
3.	Zanzibar	located in	?
4.	Zanzibar	flew to	?
5.	Dr. Robert	recommended	PER
6.	Oprah	recommended	?

Algorithm

- Use classifier 1 to label example 5.

Co-training example

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1.	Peachtree Street	located on	LOC
2.	Dr. Walker	said	PER
3.	Zanzibar	located in	LOC
4.	Zanzibar	flew to	?
5.	Dr. Robert	recommended	PER
6.	Oprah	recommended	?

Algorithm

- Use classifier 1 to label example 5.
- Use classifier 2 to label example 3.

Co-training example

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1.	Peachtree Street	located on	LOC
2.	Dr. Walker	said	PER
3.	Zanzibar	located in	LOC
4.	Zanzibar	flew to	?
5.	Dr. Robert	recommended	PER
6.	Oprah	recommended	?

Algorithm

- Use classifier 1 to label example 5.
- Use classifier 2 to label example 3.
- Retrain both classifiers, using newly labeled data.

Co-training example

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1.	Peachtree Street	located on	LOC
2.	Dr. Walker	said	PER
3.	Zanzibar	located in	LOC
4.	Zanzibar	flew to	LOC
5.	Dr. Robert	recommended	PER
6.	Oprah	recommended	PER

Algorithm

- Use classifier 1 to label example 5.
- Use classifier 2 to label example 3.
- Retrain both classifiers, using newly labeled data.
- Use classifier 1 to label example 4.
- Use classifier 2 to label example 6.

Building a graph of related instances

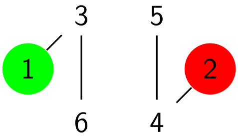
Back to sentiment analysis in French...

1. ☺ émouvant avec grâce et style
2. ☹ fastidieusement inauthentique et banale
3. pleine de style et intrigue
4. la banalité n'est dépassée que par sa prétention
5. prétentieux, de la première minute au rideau final
6. imprégné d'un air d'intrigue

Building a graph of related instances

Back to sentiment analysis in French...

1. ☺ émouvant avec grâce et style
2. ☹ fastidieusement inauthentique et banale
3. pleine de style et intrigue
4. la banalité n'est dépassée que par sa prétention
5. prétentieux, de la première minute au rideau final
6. imprégné d'un air d'intrigue



- We can view this data as a **graph**, with edges between similar instances.
- Unlabeled instances propagate information through the graph.

Graphs over instances

- Often we compute similarity from features,

$$\mathrm{sim}(i,j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

and build an edge between i and j when $\mathrm{sim}(i,j) > \tau$

Graphs over instances

- Often we compute similarity from features,

$$\text{sim}(i,j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

  and build an edge between i and j when $\text{sim}(i,j) > \tau$
- But sometimes there is a natural similarity metric.
  - For example, Pang and Lee (2004) use proximity in the document for subjectivity analysis.
  - The idea is that adjacent sentences are more likely to have the same subjectivity status.

Minimum cuts

Pang and Lee use **minimum cuts** to assign subjectivity in a proximity graph of sentences.

$$
\begin{aligned}
y_i &\in \{0, 1\} \\
\text{Fix } Y_l &= \{y_1, y_2, \ldots y_\ell\} \\
\text{Solve for } Y_u &= \{y_{\ell+1}, \ldots, y_{\ell+m}\} \\
\min_{Y_u} &\sum_{i,j} w_{ij}(y_i - y_j)^2
\end{aligned}
$$

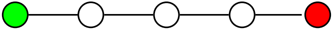
Minimum cuts

Pang and Lee use **minimum cuts** to assign subjectivity in a proximity graph of sentences.

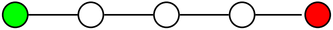
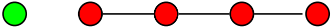
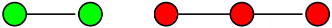
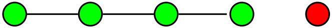
$$
\begin{aligned}
y_i &\in \{0, 1\} \\
\text{Fix } Y_l &= \{y_1, y_2, \ldots y_\ell\} \\
\text{Solve for } Y_u &= \{y_{\ell+1}, \ldots, y_{\ell+m}\} \\
&\min_{Y_u} \sum_{i,j} w_{ij}(y_i - y_j)^2
\end{aligned}
$$

- This looks like a combinatorial problem...
- But assuming $w_{ij} \geq 0$, it can be solved with maximum-flow.

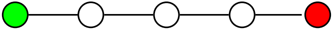
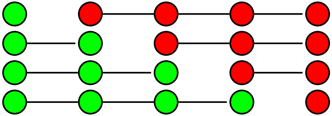
Problems with minimum cuts

- Mincuts may have several possible solutions:
  - Initial graph 

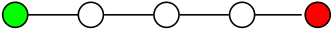
Problems with minimum cuts

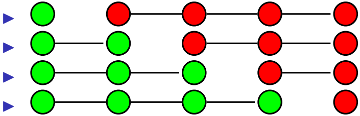
- Mincuts may have several possible solutions:
  - Initial graph 
  - Equivalent solutions
    - 
    - 
    - 
    - 

Problems with minimum cuts

- Mincuts may have several possible solutions:
  - Initial graph 
  - Equivalent solutions
    
- Another problem is that mincuts doesn't distinguish high confidence predictions.

Problems with minimum cuts

- Mincuts may have several possible solutions:
  - Initial graph 
  - Equivalent solutions



- Another problem is that mincuts doesn't distinguish high confidence predictions.
- One solution is randomized mincuts (Blum et al, 2004)
  - Add random noise to adjacency matrix.
  - Rerun mincuts multiple times.
  - Deduce the final classification by voting.

Label propagation

- Relax y_i from $\{0, 1\}$ to $\mathbb{R}$
- Minimize $\sum_{i,j} w_{ij}(y_i - y_j)^2$
- Advantages:
  - unique global optimum
  - natural notion of confidence: distance of y_i from 0.5

Label propagation on the graph Laplacian

- Let $\mathbf{W}$ be the $n \times n$ weight matrix.
- Let $\mathbf{D}$ be the **degree matrix**, $d_{ii} = \sum_j w_{ij}$. $\mathbf{D}$ is diagonal.
- The unnormalized **graph Laplacian** is $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- We want to minimize the energy $\sum_i w_{ij}(y_i - y_j)^2 = \mathbf{y}^\mathsf{T}\mathbf{L}\mathbf{y}$, subject to the constraint that we can't change $\mathbf{y}_\ell$.

Label propagation on the graph Laplacian

- Let $\mathbf{W}$ be the $n \times n$ weight matrix.
- Let $\mathbf{D}$ be the **degree matrix**, $d_{ii} = \sum_j w_{ij}$. $\mathbf{D}$ is diagonal.
- The unnormalized **graph Laplacian** is $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- We want to minimize the energy $\sum_i w_{ij}(y_i - y_j)^2 = \mathbf{y}^\mathsf{T}\mathbf{L}\mathbf{y}$, subject to the constraint that we can't change $\mathbf{y}_\ell$.
- Solution:
  - Partition the Laplacian $\mathbf{L} = \begin{bmatrix} \mathbf{L}_{\ell\ell} & \mathbf{L}_{\ell u} \\ \mathbf{L}_{u\ell} & \mathbf{L}_{uu} \end{bmatrix}$
  - Then the closed form solution is $\mathbf{y}_u = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{u\ell}\mathbf{y}_\ell$
  - This is great ... if we can invert $\mathbf{L}_{uu}$.

Iterative label propagation

- $\mathbf{L}_{u,u}$ is huge, so we can't invert it unless it has special structure.
- Iterative solution from Zhu and Ghahramani (2002):
  - Let $\mathbf{T}_{ij} = \frac{w_{ij}}{\sum_k w_{kj}}$, row-normalizing $\mathbf{W}$.
  - Let $\mathbf{Y}$ be an $n \times C$ matrix of labels, where C is the number of classes. In the R&R reading, a special "default" label is used for the unlabeled nodes.
  - Until tired,
    - Set $\mathbf{Y} = \mathbf{TY}$
    - Row-normalize $\mathbf{Y}$
    - Clamp the seed examples in $\mathbf{Y}$ to their original values

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?
- Manifold regularization (Belkin et al 2006) favors learning models that are **smooth** over the similarity graph.
  - we want to learn a function $f : \mathcal{X} \to \mathcal{Y}$ which is:
    - correct on the labeled data
    - simple (regularized)
    - smooth on the graph (or manifold)

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?
- Manifold regularization (Belkin et al 2006) favors learning models that are **smooth** over the similarity graph.
  - we want to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which is:
    - correct on the labeled data
    - simple (regularized)
    - smooth on the graph (or manifold)

$$\arg\min_f \frac{1}{\ell} \sum_i \ell(f(\mathbf{x}_i), y_i) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?
- Manifold regularization (Belkin et al 2006) favors learning models that are **smooth** over the similarity graph.
  - we want to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which is:
    - **correct on the labeled data**
    - simple (regularized)
    - smooth on the graph (or manifold)

$$\arg\min_{f} \frac{1}{\ell} \sum_{i} \ell(f(\mathbf{x}_i), y_i) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?
- Manifold regularization (Belkin et al 2006) favors learning models that are **smooth** over the similarity graph.
  - we want to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which is:
    - correct on the labeled data
    - **simple (regularized)**
    - smooth on the graph (or manifold)

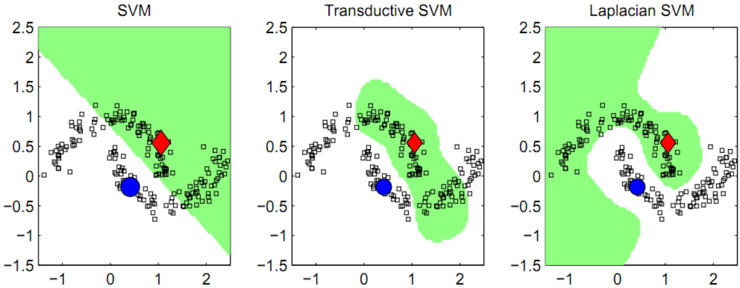
$$\arg\min_f \frac{1}{\ell} \sum_i \ell(f(\mathbf{x}_i), y_i) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j} w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

Manifold regularization

- Graph propagation is **transductive** learning.
  - It learns by a joint operation on the labeled and unlabeled data.
  - What if new test data arrives later?
- Manifold regularization (Belkin et al 2006) favors learning models that are **smooth** over the similarity graph.
  - we want to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which is:
    - correct on the labeled data
    - simple (regularized)
    - **smooth on the graph (or manifold)**

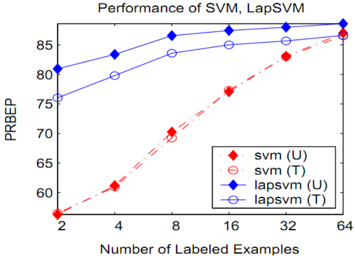
$$\arg\min_{f} \frac{1}{\ell} \sum_{i} \ell(f(\mathbf{x}_i), y_i) + \lambda_1 ||f||^2 + \lambda_2 \sum_{i,j} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$$

Manifold regularization: synthetic data



Manifold regularization: text classification

- Text classification: mac versus $\mathrm{windows}$
- Each document is represented by its TF-IDF vector
- The graph W is constructed from 15-nearest-neighbors (in TF-IDF space)



How can we learn with less annotation effort?

- **Semisupervised learning**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$: labeled examples
  - $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$: unlabeled examples
  - often $u \gg \ell$

How can we learn with less annotation effort?

- **Semisupervised learning**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$: labeled examples
  - $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$: unlabeled examples
  - often $u \gg \ell$

- **Domain adaptation**
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$: labeled examples in *source* domain
  - $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_T} \sim \mathcal{D}_T$: labeled examples in *target* domain
  - possibly some unlabeled data in target and possibly source domain
  - evaluate in the target domain

The current status of NER

Quote from Wikipedia

"State-of-the-art NER systems produce near-human performance. For example, the best system entering MUC-7 scored 93.39% of f-measure while human annotators scored 97.60% and 96.95%"

The current status of NER

Quote from Wikipedia

"State-of-the-art NER systems produce near-human performance. For example, the best system entering MUC-7 scored 93.39% of f-measure while human annotators scored 97.60% and 96.95%"

Wow, that is so cool! At the end, we finally solved something!

The current status of NER

Quote from Wikipedia

"State-of-the-art NER systems produce near-human performance. For example, the best system entering MUC-7 scored 93.39% of f-measure while human annotators scored 97.60% and 96.95%"

Wow, that is so cool! At the end, we finally solved something!

Truth: The NER problem is still not solved. Why?

The problem: domain over-fitting

- The issues of supervised machine learning algorithms:
  Need Labeled Data
- What people have done: Labeled large amount of data on news corpus
- However, it is still not enough.....
- The Web contains all kind of data....
  - Blogs, Novels, Biomedical Documents, . . .
  - Many domains!
- We might do a good job on news domain, but not on other domains...

Domain Adaptation

- Many NLP tasks are cast into classification problems
- Lack of training data in new domains
- Domain adaptation:
  - POS: WSJ → biomedical text
  - NER: news → blog, speech
  - Spam filtering: public email corpus → personal inboxes
- Domain overfitting

NER Task	Train → Test	F1
to find PER, LOC, ORG from news text	NYT → NYT	0.855
	Reuters → NYT	0.641
to find gene/protein from biomedical literature	mouse → mouse	0.541
	fly → mouse	0.281

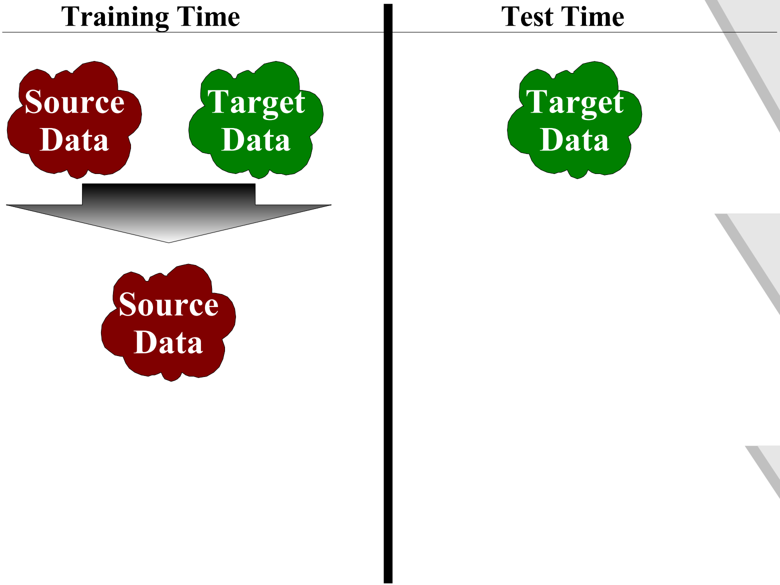
Supervised domain adaptation

In supervised domain adaptation, we have:

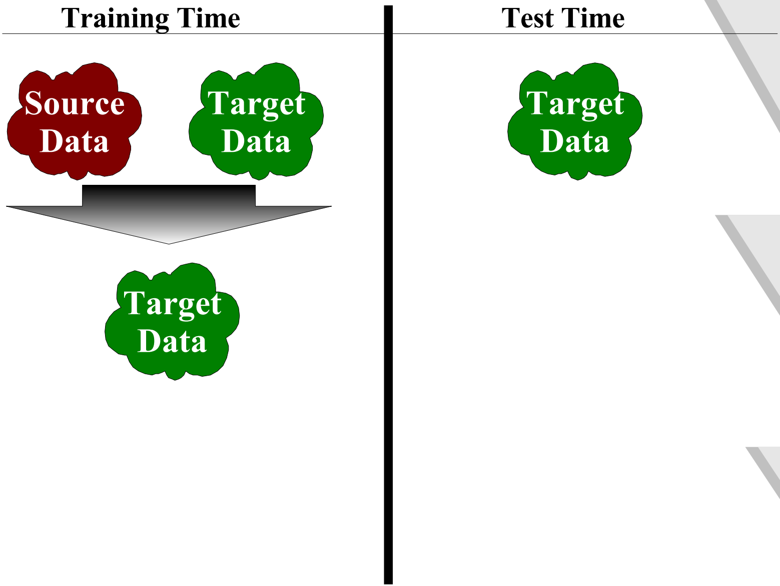
- Lots of labeled data in a "source" domain, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$ (e.g., reviews of restaurants)

- A little labeled data in a "target" domain, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_T} \sim \mathcal{D}_T$ (e.g., reviews of chess stores)

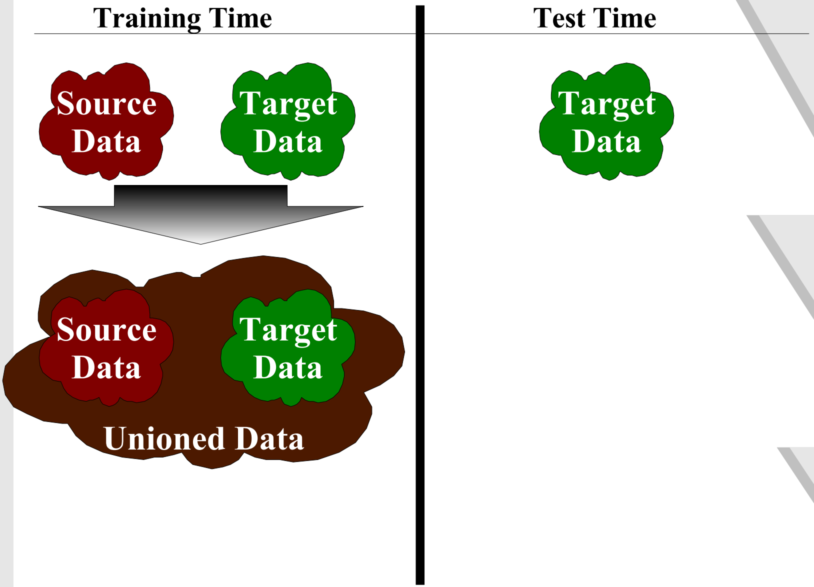
Obvious Approach 1: SrcOnly



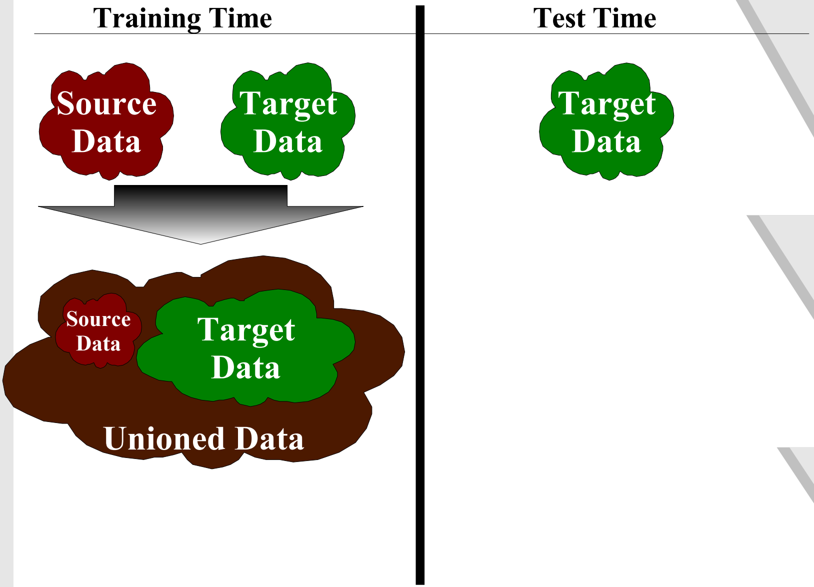
Obvious Approach 2: TgtOnly



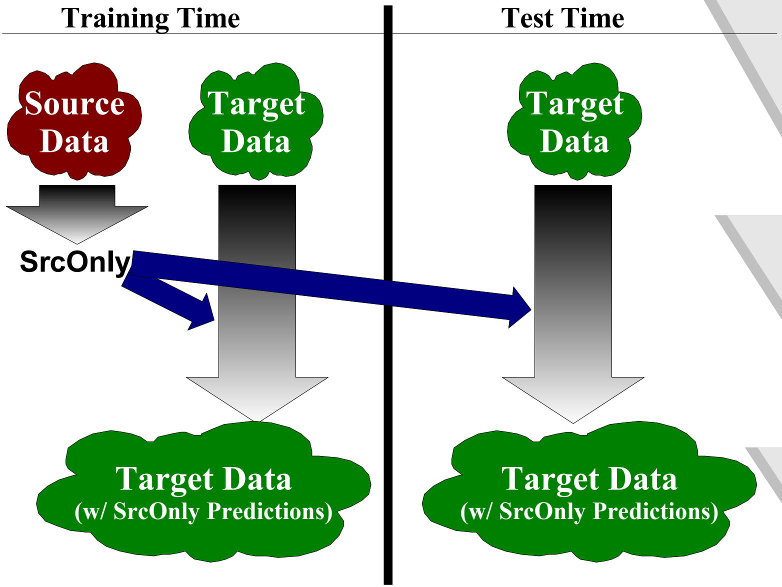
Obvious Approach 3: All



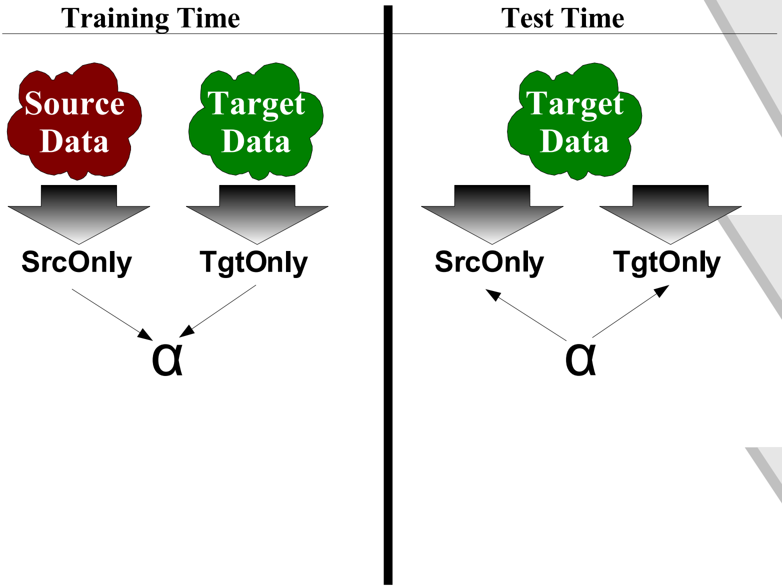
Obvious Approach 4: Weighted



Obvious Approach 5: Pred



Obvious Approach 6: LinInt



Less obvious approaches

- **Priors** (Chelba and Acero 2004)
  - Let $\mathbf{w}^{(S)}$ be the optimal weights in the source domain.
  - Design a prior distribution $P(\mathbf{w}^{(T)}|\mathbf{w}^{(S)})$
  - Solve $\mathbf{w}^{(T)} = \arg\max_{\mathbf{w}} \log P(\mathbf{y}^{(T)}|\mathbf{x}^{(T)}) + \log P(\mathbf{w}^{(T)}|\mathbf{w}^{(S)})$

Less obvious approaches

- **Priors** (Chelba and Acero 2004)
  - Let $\mathbf{w}^{(S)}$ be the optimal weights in the source domain.
  - Design a prior distribution $P(\mathbf{w}^{(T)}|\mathbf{w}^{(S)})$
  - Solve $\mathbf{w}^{(T)} = \arg\max_{\mathbf{w}} \log P(\mathbf{y}^{(T)}|\mathbf{x}^{(T)}) + \log P(\mathbf{w}^{(T)}|\mathbf{w}^{(S)})$
- **Feature augmentation** (Daume III 2007)

"MONITOR" versus "THE"

News domain:
"MONITOR" is a **verb**
"THE" is a **determiner**

Technical domain:
"MONITOR" is a **noun**
"THE" is a **determiner**

Key Idea:
Share some features ("the")
Don't share others ("monitor")

(and let the *learner* decide which are which)

Feature Augmentation

	We	monitor	the	traffic	The	monitor	is	heavy
	N	**V**	**D**	**N**	**D**	**N**	**V**	**R**
Original Features		W:monitor P:we N:the C:a+	W:the P:monitor N:traffic C:a+		W:the P:<s> N:monitor C:Aa+	W:monitor P:the N:is C:a+		
Augmented Features		SW:monitor SP:we SN:the SC:a+	SW:the SP:monitor SN:traffic SC:a+		TW:the TP:<s>	TW:monitor TP:the		

Why should this work?

In feature-vector lingo:

$\Phi(x) \rightarrow \langle \Phi(x), \Phi(x), 0 \rangle$ (for source domain)

$\Phi(x) \rightarrow \langle \Phi(x), 0, \Phi(x) \rangle$ (for target domain)

Results – Error Rates

Task	Dom	SrcOnly	TgtOnly	Baseline	Prior	Augment
ACE-NER	bn	4.98	2.37	2.11 (pred)	2.06	**1.98**
	bc	4.54	4.07	3.53 (weight)	**3.47**	**3.47**
	nw	4.78	3.71	3.56 (pred)	3.68	**3.39**
	wl	2.45	2.45	**2.12** (all)	2.41	**2.12**
	un	3.67	2.46	2.10 (linint)	2.03	**1.91**
	cts	2.08	0.46	0.40 (all)	**0.34**	**0.32**
CoNLL	tgt	2.49	2.95	**1.75** (wgt/li)	1.89	**1.76**
PubMed	tgt	12.02	4.15	3.95 (linint)	3.99	**3.61**
CNN	tgt	10.29	3.82	3.44 (linint)	**3.35**	**3.37**
Tree bank-Chunk	wsj	6.63	4.35	4.30 (weight)	4.27	**4.11**
	swbd3	15.90	4.15	4.09 (linint)	3.60	**3.51**
	br-cf	5.16	6.27	**4.72** (linint)	5.22	5.15
	br-cg	4.32	5.36	**4.15** (all)	4.25	4.90
	br-ck	5.05	6.32	**5.01** (prd/li)	5.27	5.41
	br-cl	5.66	6.60	**5.39** (wgt/prd)	5.99	5.73
	br-cm	3.57	6.59	**3.11** (all)	4.08	4.89
	br-cn	4.60	5.56	**4.19** (prd/li)	4.48	4.42
	br-cp	4.82	5.62	**4.55** (wgt/prd/li)	4.87	4.78
	br-cr	5.78	9.13	**5.15** (linint)	6.71	6.30
Treebank-	brown	6.35	5.75	4.72 (linint)	4.72	**4.65**

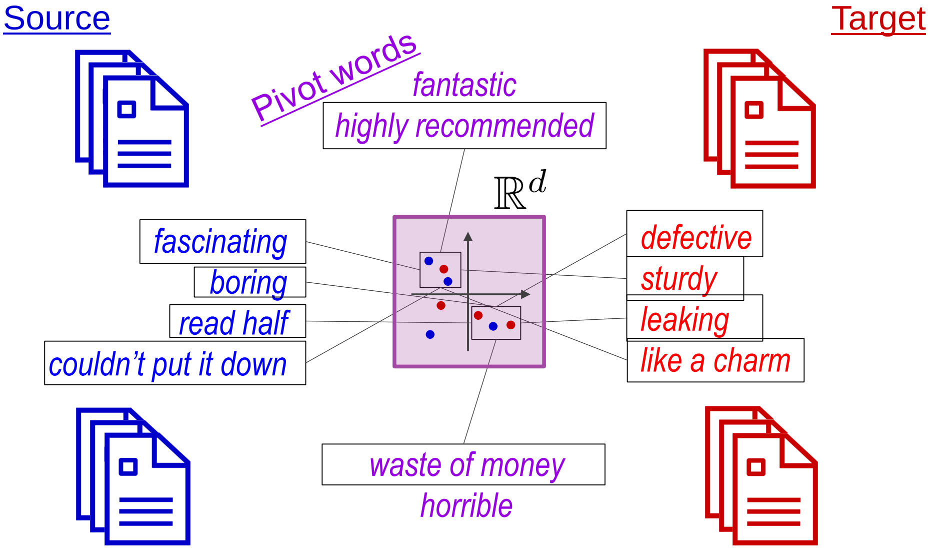
Unsupervised domain adaptation

In unsupervised domain adaptation, we have:

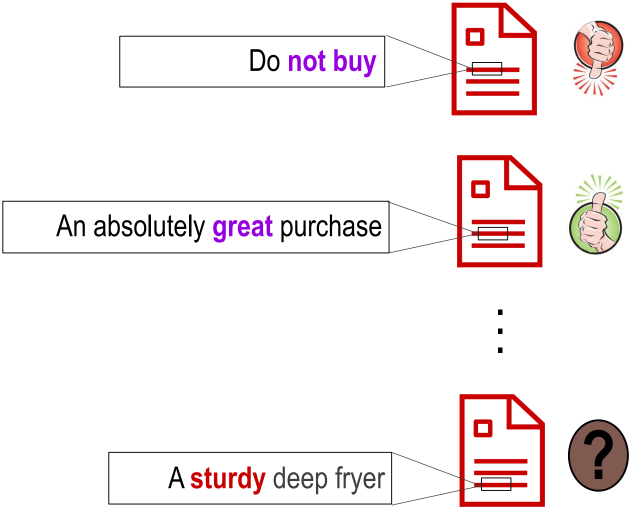
- Lots of labeled data in a "source" domain, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$ (e.g., reviews of restaurants)

- Lots of unlabeled data in a "target" domain, $\{(\mathbf{x}_i)\}_{i=1}^{\ell_T} \sim \mathcal{D}_T$ (e.g., reviews of chess stores)

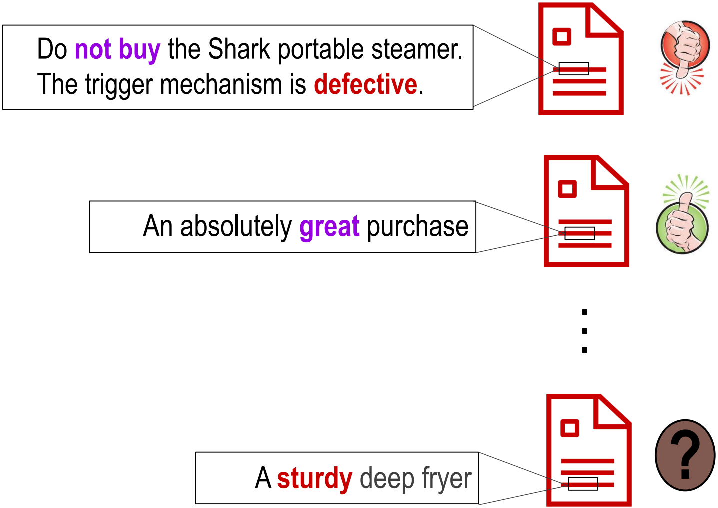
Learning Representations: Pivots



Predicting pivot word presence



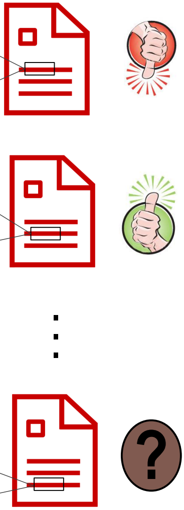
Predicting pivot word presence



Predicting pivot word presence

Do **not buy** the Shark portable steamer. The trigger mechanism is **defective**.

An absolutely **great** purchase. . . . This blender is incredibly **sturdy**.



Predict presence of pivot words

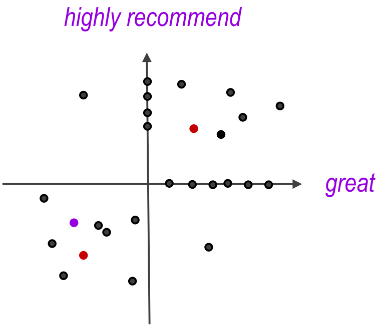
$$p_{w(\textit{great})}(\textit{great}\,|\,x) \propto \exp\{\langle x, w(\textit{great})\rangle\}$$

A **sturdy** deep fryer

Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

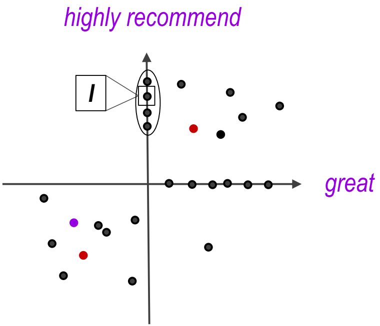
- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: “Did *highly recommend* appear?”
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

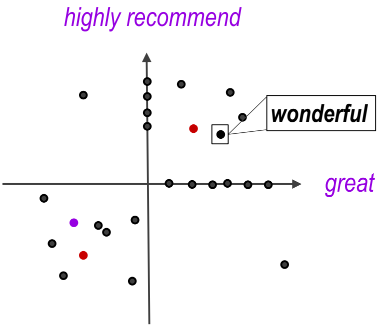
- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \ldots & w(\substack{\text{highly} \\ \text{recommend}}) & \ldots & w_N \\ | & & | & & | \end{bmatrix}$$

- $p_W(\textit{pivots} | x)$ generates N new features
- $p_{w(\substack{\text{highly} \\ \text{recommend}})}(\substack{\text{highly} \\ \text{recommend}} | x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



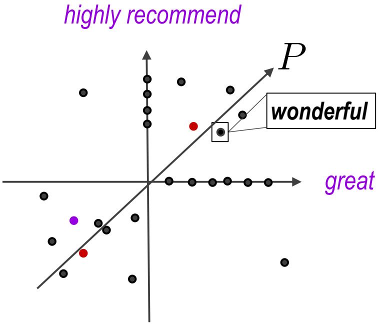
Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\text{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

- Let P be a basis for the subspace of best fit to W

- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\text{highly recommend})}(\text{highly recommend}|x)$: "Did *highly recommend* appear?"

- Sometimes predictors capture non-sentiment information



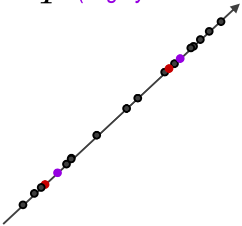
Finding a shared sentiment subspace

$$W = \begin{bmatrix} | & & | & & | \\ w_1 & \dots & w(\textit{highly recommend}) & \dots & w_N \\ | & & | & & | \end{bmatrix}$$

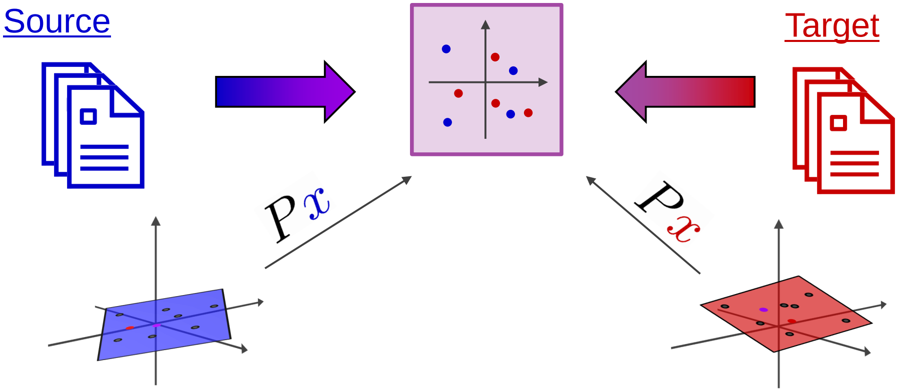
- Let P be a basis for the subspace of best fit to W
- P captures sentiment variance in W

- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information

P (*highly recommend, great*)

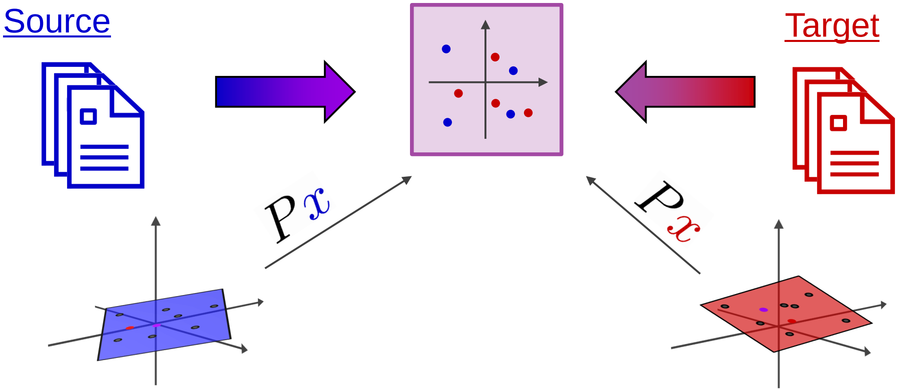


P projects onto shared subspace



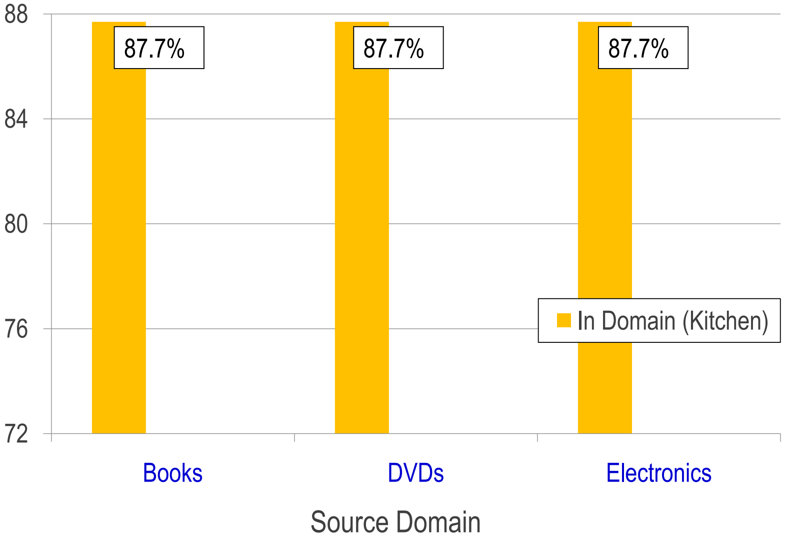
$$p_{\tilde{\theta}}(\quad|x) \propto \exp\left\{\langle \phi(\quad, Px), \tilde{\theta}\rangle\right\}$$

P projects onto shared subspace

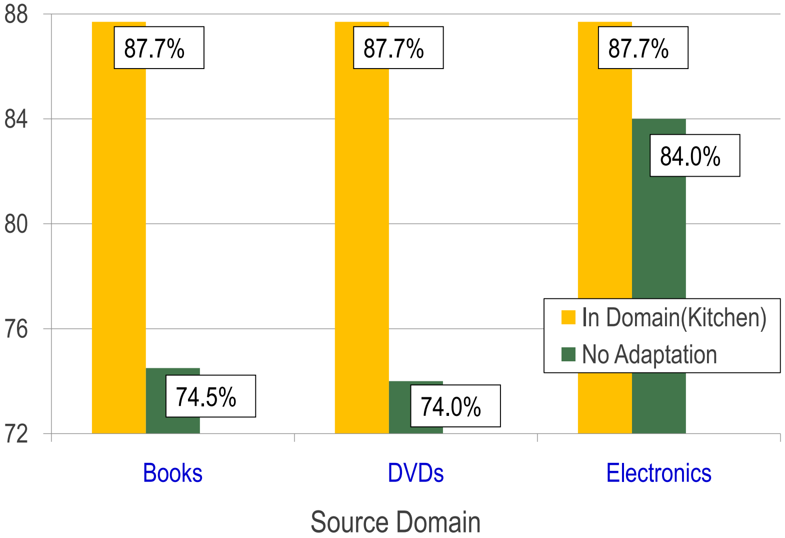


$$h(x) = \mathrm{sgn}\left(\theta^\top P x\right)$$

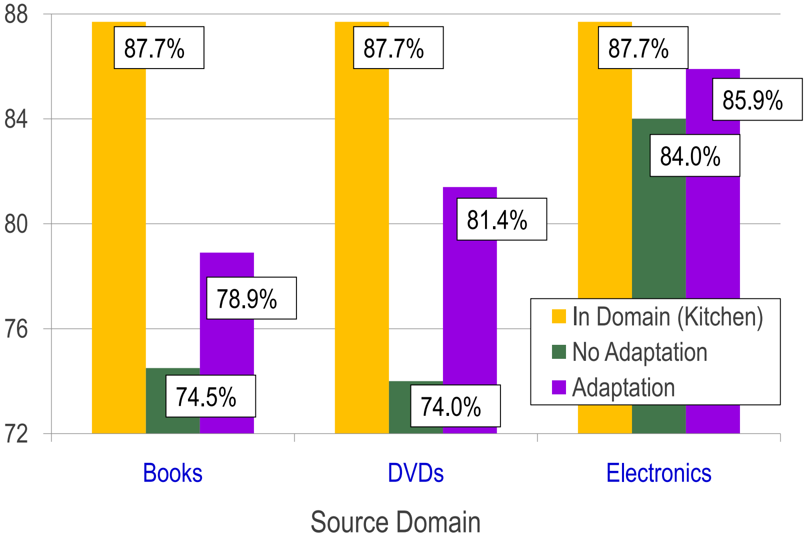
Target Accuracy: Kitchen Appliances



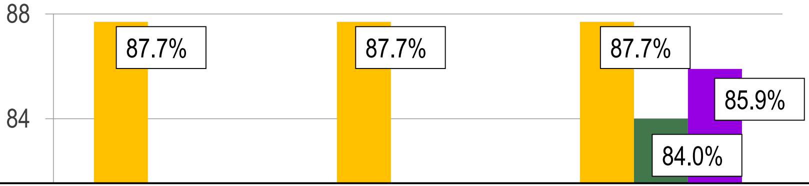
Target Accuracy: Kitchen Appliances



Target Accuracy: Kitchen Appliances



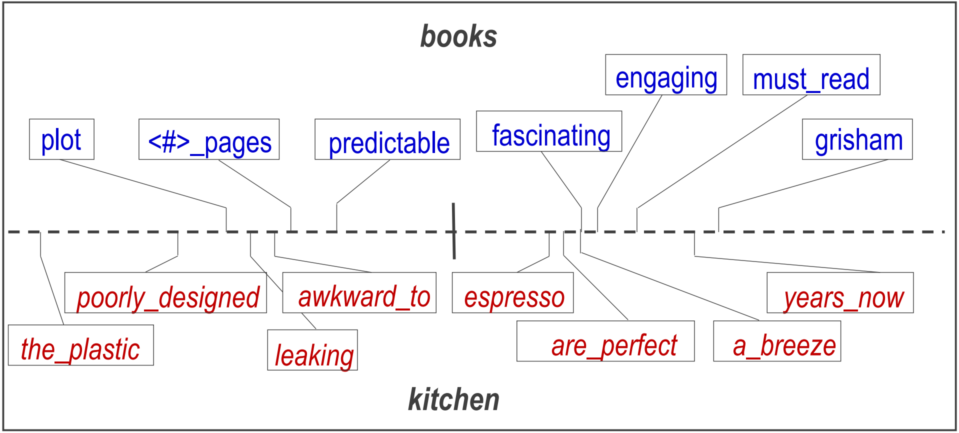
Adaptation Error Reduction



<u>36% reduction in error due to adaptation</u>

Visualizing P (books & kitchen)

negative vs. positive

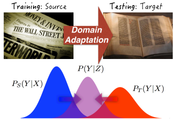


Fast Easy Unsupervised Domain Adaptation with Marginalized Structured Dropout

Yi Yang and Jacob Eisenstein

Georgia Tech

Domain Adaptation



Example: Part-of-speech Tagging

Source:

And God said, Let …
CC NNP VBD, VB …

Features:

- Mid_said source spec
- Prev_God cross domain
- Next_Let source spec
- …

Target:

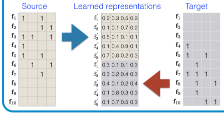
And God seide, Liyt …
CC NNP VBD, VB …

Features:

- Mid_seide target spec
- Prev_God cross domain
- Next_Liyt target spec
- …

Representation Learning

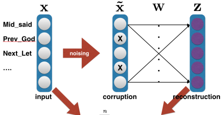
Learn new sets of dense features:



Representation learning for domain adaptation:

- *Pivots*: Structural Corresponding Learning (SCL) [1]
- *Clustering*: Brown Clustering
- *Latent Variable Models*: Topic Model, Hidden Markov Model
- *Deep Learning*: (marginalized) Stacked Denoising Autoencoders (SDA/mSDA) [2,3]

Denoising Autoencoders



$$\mathcal{L} = \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i||^2$$

Reconstruct 'pivots': $\mathcal{L} = \sum_{i=1}^{n} ||\mathbf{x}_i^{pivot} - \mathbf{W}\tilde{\mathbf{x}}_i||^2$

Closed-form solution: $\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1}$, with $\mathbf{P} = \sum_{i=1}^{n} \mathbf{x}_i\tilde{\mathbf{x}}_i^\top$ and $\mathbf{Q} = \sum_{i=1}^{n} \tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top$

Marginalized Denoising Autoencoders []: $\mathbf{P} = \sum_{i=1}^{n} E[\mathbf{x}_i\tilde{\mathbf{x}}_i^\top]$, and $\mathbf{Q} = \sum_{i=1}^{n} E[\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top]$

Learned representations: $\tanh(\mathbf{W}\mathbf{X})$

Compute P and Q under dropout noise: For each feature of an instance, remove it with probability p.

$$\mathbf{Q}_{\alpha,\beta} = \begin{cases} (1-p)^2\mathbf{S}_{\alpha,\beta} & \text{if } \alpha \neq \beta \\ (1-p)\mathbf{S}_{\alpha,\beta} & \text{if } \alpha = \beta \end{cases},$$

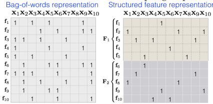
$$\mathbf{P}_{\alpha,\beta} = (1-p)\mathbf{S}_{\alpha,\beta},$$

where $\mathbf{S} = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$ is the scatter matrix, α and β index two features.

Structured Dropout

Bag-of-words (BoW) vs. structured feature representations:

- *Bag-of-words*: features fire anywhere.
- *Structured*: one feature fires per group.



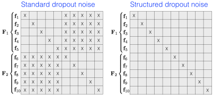
Compute P and Q under structured dropout: Randomly choose one active feature (type) to keep, dropout all other features.

$$\mathbf{Q}_{\alpha,\beta} = \begin{cases} 0 & \text{if } \alpha \neq \beta \\ \frac{1}{K}\mathbf{S}_{\alpha,\beta} & \text{if } \alpha = \beta \end{cases}$$

$$\mathbf{P}_{\alpha,\beta} = \frac{1}{K}\mathbf{S}_{\alpha,\beta},$$

where K is the number of feature types. There is no free hyperparameter.

Shape of Q under different noises:



Evaluation

Datasets: Tycho Brahe corpus (historical Portuguese texts with 383 tags)

Dataset	# of Tokens				
	Total	Narrative	Letters	Dissertation	Theatre
1800-1849	125719	91582	34137	0	0
1750-1799	202346	57477	84465	0	60404
1700-1749	278846	0	130327	148519	0
1650-1699	248194	83938	115062	49194	0
1600-1649	295154	117515	115252	62387	0
1550-1599	148061	148061	0	0	0
1500-1549	182208	126516	0	55692	0
Overall	1480528	625089	479243	315792	60404

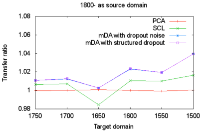
Experiment setup:

- *CRF tagger*: 16 feature types, 372,902 features, and 1572 pivots.
- *Methods*: baseline, PCA, SCL
- *Parameters*: decided with development data on the training set.

Representation learning time:

Method	PCA	SCL	mDA	
			dropout	structured
Time (sec)	7,779	38,849	8,939	**339**

Results:



Task	baseline	PCA	SCL	mDA	
				dropout	structured
from 1800-1849					
→ 1750	89.12	89.09	89.69	**90.08**	**90.08**
→ 1700	90.43	90.43	91.06	91.56	**91.57**
→ 1650	88.45	88.52	87.09	88.69	**88.70**
→ 1600	87.56	87.58	88.47	89.60	**89.61**
→ 1550	89.66	89.61	90.57	**91.39**	**91.39**
→ 1500	85.58	85.63	86.99	**88.96**	88.95
from 1750-1849					
→ 1700	94.64	94.62	94.81	**95.08**	**95.08**
→ 1650	**91.98**	90.97	90.37	90.83	90.84
→ 1600	92.95	92.91	93.17	**93.78**	**93.78**
→ 1550	93.27	93.21	93.75	**94.06**	94.05
→ 1500	89.80	89.75	90.59	**91.71**	**91.71**

References

[1] John Blitzer et al. Domain Adaptation with Structural Correspondence Learning. In *EMNLP'06*.

[2] Xavier Glorot et al. Domain Adaptation for Large-Scale Sentiment ClassiñĄcation: A Deep Learning Approach. In *ICML '11*

[3] Minmin Chen et al. Marginalized Denoising Autoencoders for Domain Adaptation. In *ICML '12*

Acknowledgments

This research was supported by National Science Foundation award 1349837. The first author was also supported by National Science Foundation ACL travel award.

Recap

- In application settings,
  - You rarely have all the labeled data you want.
  - You often have lots of unlabeled data.
- Semi-supervised learning learns from unlabeled data too:
  - Bootstrapping (or self-training) works best when you have multiple orthogonal views: for example, string and context.
  - Probabilistic methods *impute* the labels of unseen data.
  - Graph-based methods encourage similar instances or types to have similar labels.

Alternative frameworks

- **Semisupervised learning**
  - learn from labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$
  - and unlabeled examples $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$
  - often $u \gg \ell$

Alternative frameworks

- **Semisupervised learning**
  - learn from labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$
  - and unlabeled examples $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$
  - often $u \gg \ell$
- **Domain adaptation**
  - learn from lots of labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_S$ in a *source* domain
  - learn from a few (or zero) labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_T$ in a *target* domain
  - evaluate in the target domain

Alternative frameworks

- **Semisupervised learning**
  - learn from labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$
  - and unlabeled examples $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$
  - often $u \gg \ell$
- **Domain adaptation**
  - learn from lots of labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_S$ in a *source* domain
  - learn from a few (or zero) labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_T$ in a *target* domain
  - evaluate in the target domain
- **Active learning**: model can query annotator for labels

Alternative frameworks

- **Semisupervised learning**
  - learn from labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell}$
  - and unlabeled examples $\{(\mathbf{x}_i)\}_{i=\ell+1}^{\ell+u}$
  - often $u \gg \ell$
- **Domain adaptation**
  - learn from lots of labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_S$ in a *source* domain
  - learn from a few (or zero) labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell} \sim \mathcal{D}_T$ in a *target* domain
  - evaluate in the target domain
- **Active learning**: model can query annotator for labels
- **Feature labeling**
  - Provide prototypes of each label (Haghighi and Klein 2006)
  - Give rough probabilistic constraints, e.g. $\mathrm{Mr.}$ preceeds a person name at least 90% of the time (Druck et al 2008)