CS 4650/7650
Semi-Supervised Learning[1]

Jacob Eisenstein

October 24, 2013

[1] With slides borrowed from John Blitzer and Xiaojin Zhu

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]

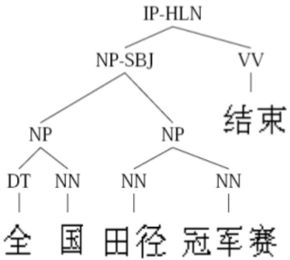
- "Switchboard" dataset of telephone conversations
- Annotations from word to phoneme sequence:
  - film → F IH_N UH_GL_N M
  - be all → BCL B IY IY_TR AO_TR AO L_DL
- **400 hours** annotation time per hour of speech!

[2] Examples from Xiaojin "Jerry" Zhu

Examples

Natural language parsing[3]

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



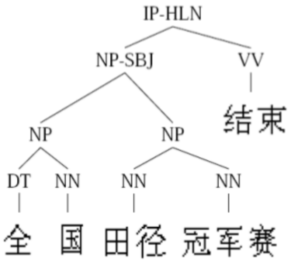
"The National Track and Field Championship has finished."

[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 d'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 d'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 d'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 d'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 d'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 d'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 d'intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'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 d'intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'intrigue

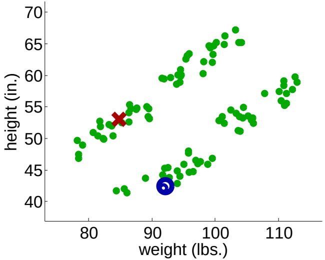
How can unlabeled data help in NLP?

Let's learn to do sentiment analysis in French.

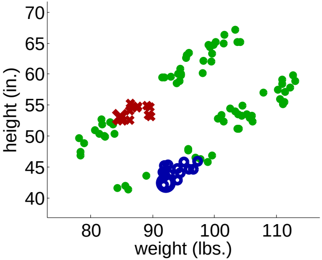
- labeled data
  - ☺ émouvant avec grâce et style
  - ☹ fastidieusement inauthentique et banale
- unlabeled data
  - pleine de style et d'intrigue
  - la banalité n'est dépassée que par sa prétention
  - prétentieux, de la première minute au rideau final
  - imprégné d'un air d'intrigue

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

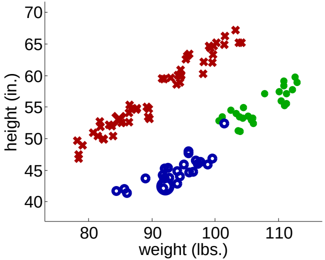
Propagating 1-Nearest-Neighbor: now it works



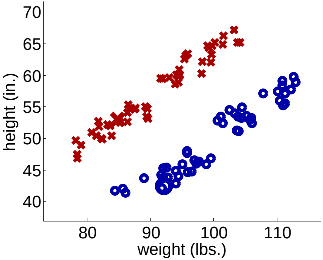
(a) Iteration 1



(b) Iteration 25



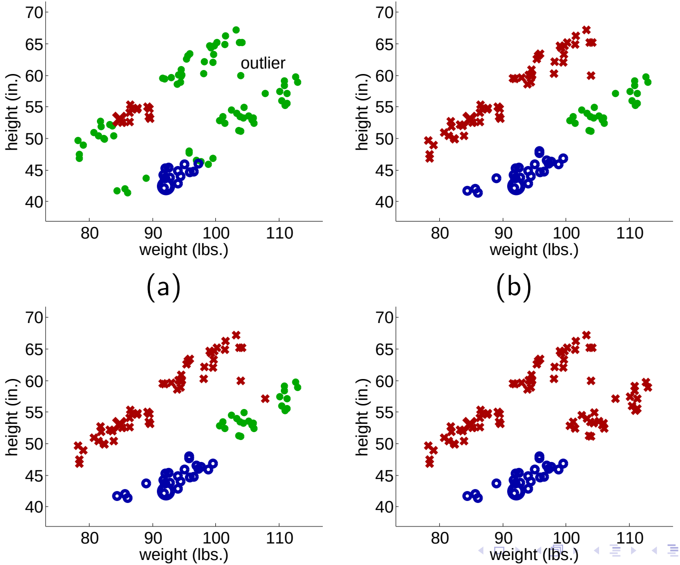
(c) Iteration 74



(d) Final labeling of all instances

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

But with a single outlier...



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

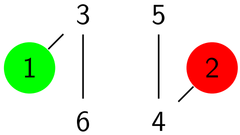
Back to sentiment analysis in French...

1. ☺ émouvant avec grâce et style
2. ☹ fastidieusement inauthentique et banale
3. pleine de style et d'¡1-¿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 d'¡1-¿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.

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.

$$y_i \in \{0, 1\}$$

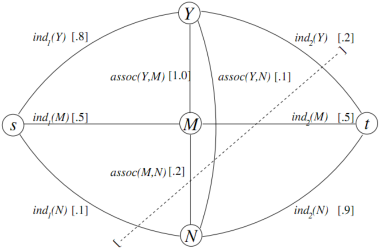
$$\text{Fix } Y_l = \{y_1, y_2, \dots y_\ell\}$$

$$\text{Solve for } Y_u = \{y_{\ell+1}, \dots, y_{\ell+m}\}$$

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

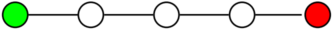
- This looks like a combinatorial problem...
- But assuming $w_{ij} \geq 0$, it can be solved with maximum-flow.

Minimum cuts for subjectivity analysis

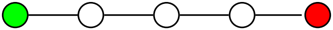
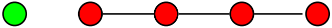
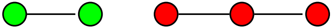
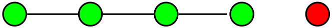


C_1	Individual penalties	Association penalties	Cost
{Y,M}	$.2 + .5 + .1$	$.1 + .2$	1.1
(none)	$.8 + .5 + .1$	0	1.4
{Y,M,N}	$.2 + .5 + .9$	0	1.6
{Y}	$.2 + .5 + .1$	$1.0 + .1$	1.9
{N}	$.8 + .5 + .9$	$.1 + .2$	2.5
{M}	$.8 + .5 + .1$	$1.0 + .2$	2.6
{Y,N}	$.2 + .5 + .9$	$1.0 + .2$	2.8
{M,N}	$.8 + .5 + .9$	$1.0 + .1$	3.3

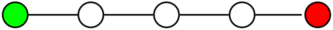
Problems with minimum cuts

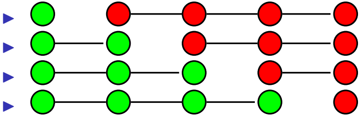
- Mincuts may have several possible solutions:
  - Initial graph 

Problems with minimum cuts

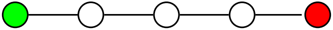
- Mincuts may have several possible solutions:
  - Initial graph 
  - Equivalent solutions
    - 
    - 
    - 
    - 

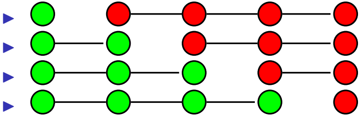
Problems with minimum cuts

- 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.

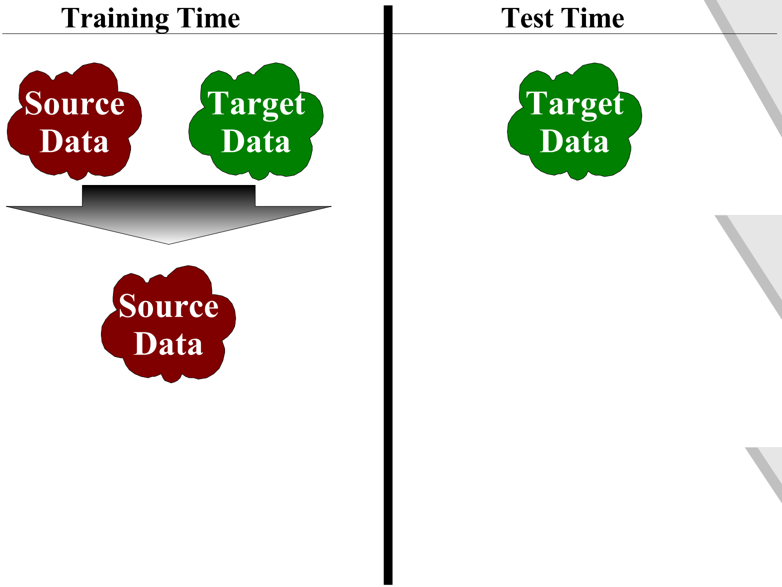
Supervised domain adaptation

In supervised domain adaptation, we have:

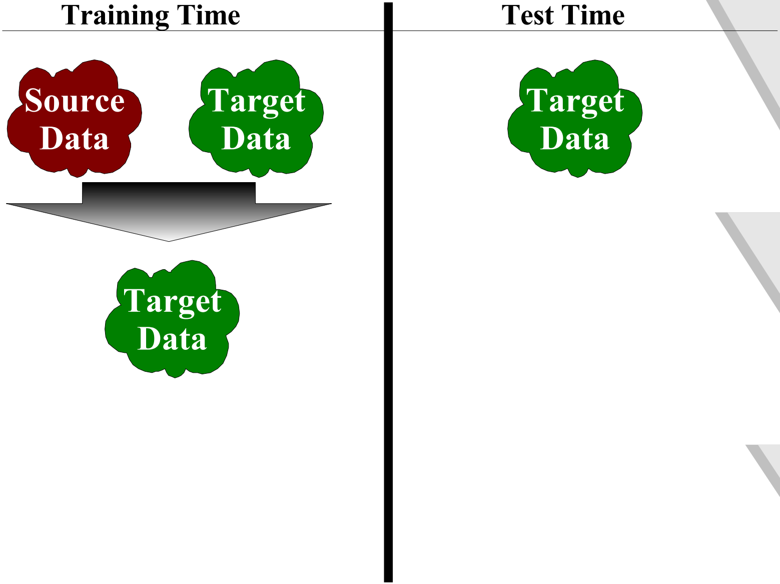
- Lots of labeled data in a "source" domain, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$ (e.g., reviews of restaurants)

- 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)

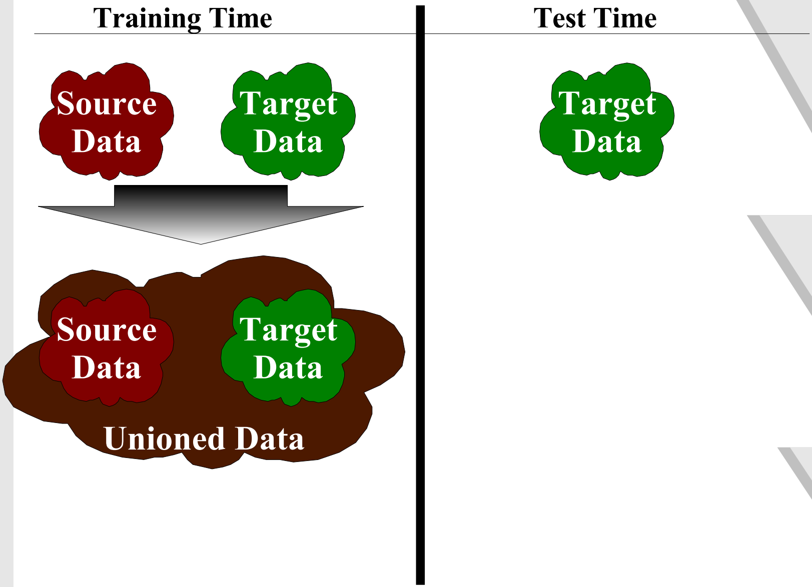
Obvious Approach 1: SrcOnly



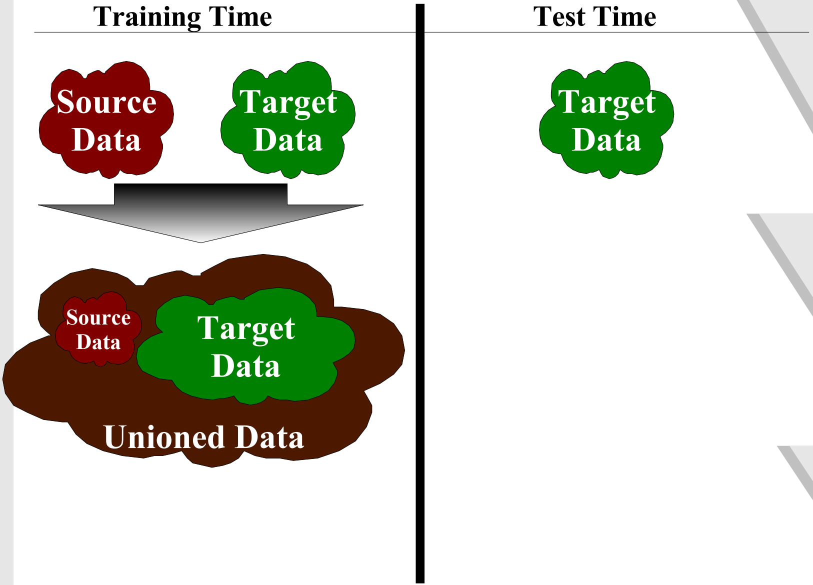
Obvious Approach 2: TgtOnly



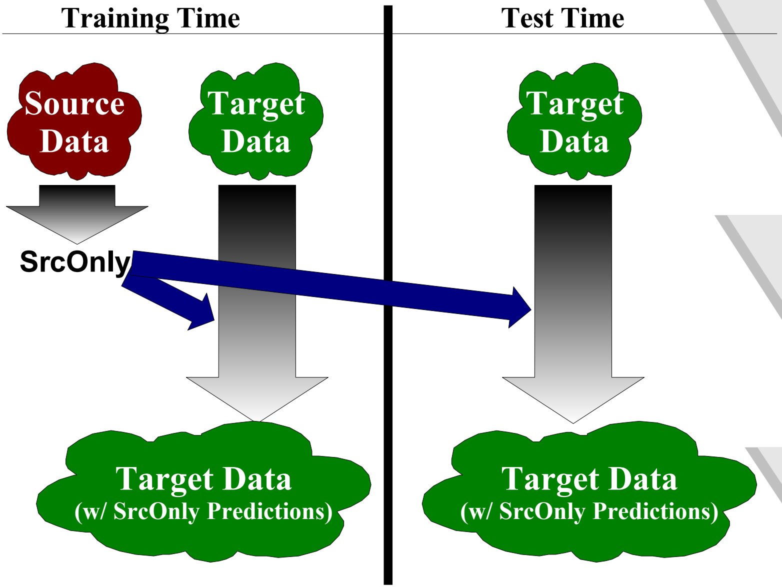
Obvious Approach 3: All



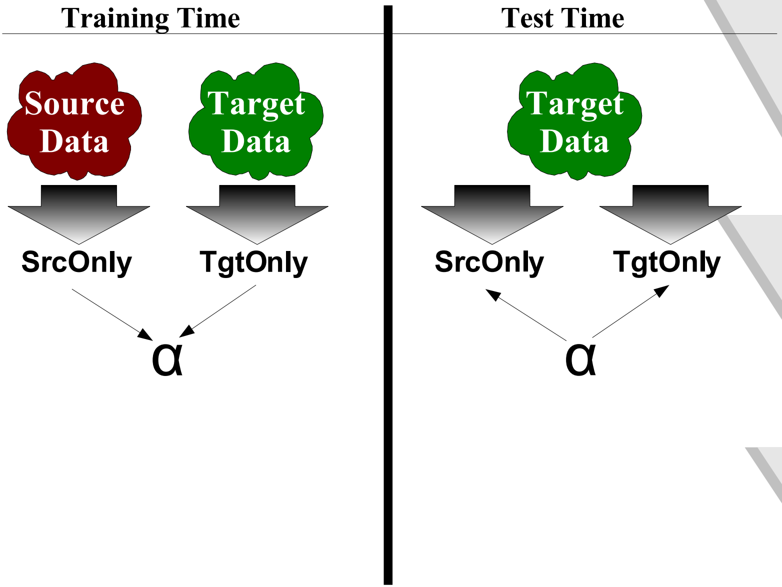
Obvious Approach 4: Weighted



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

monitor	the	The	monitor
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

monitor	the	The	monitor
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**

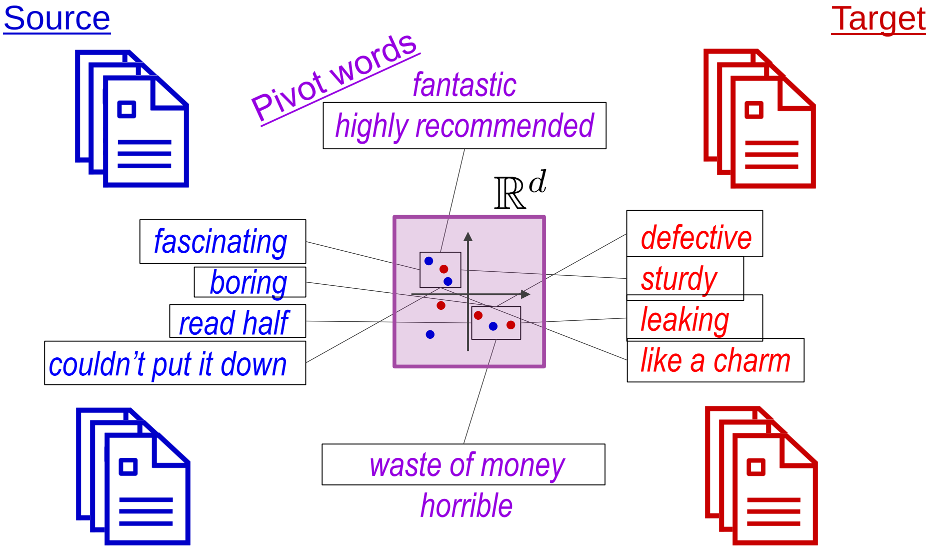
Unsupervised domain adaptation

In unsupervised domain adaptation, we have:

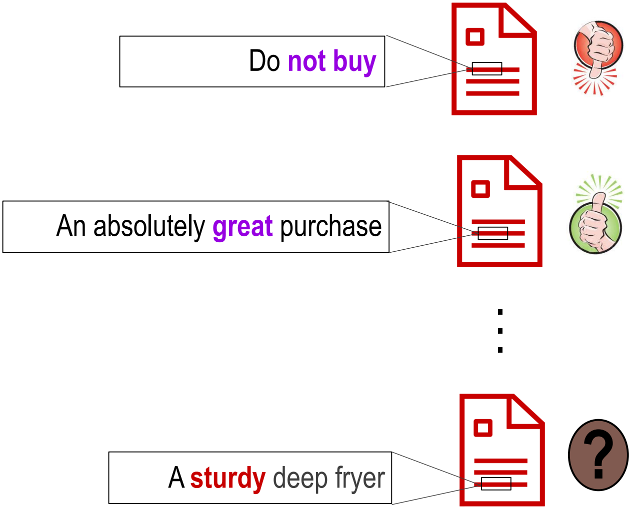
- Lots of labeled data in a "source" domain, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\ell_S} \sim \mathcal{D}_S$ (e.g., reviews of restaurants)

- 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

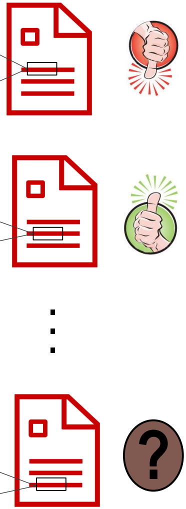


Predicting pivot word presence

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

An absolutely **great** purchase

A **sturdy** deep fryer



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

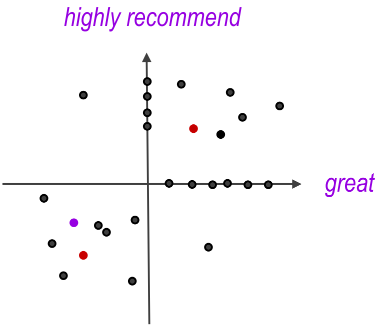
$$p_{w(\textit{great})}(\textit{great} | x) \propto \exp\{\langle x, w(\textit{great})\rangle\}$$

A **sturdy** deep fryer

Finding a shared sentiment subspace

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

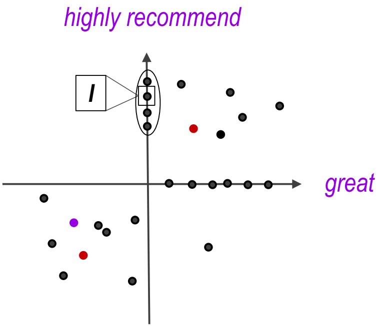
- $p_W(\textit{pivots}|x)$ generates N new features
- $p_{w(\textit{highly recommend})}(\textit{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

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

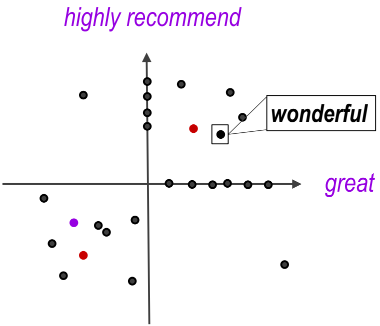
- $p_W(\text{pivots}|x)$ generates N new features
- $p_{w(\text{highly recommend})}(\text{highly recommend}|x)$: "Did *highly recommend* appear?"
- Sometimes predictors capture non-sentiment information



Finding a shared sentiment subspace

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

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

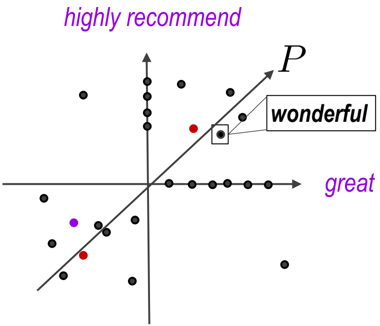


Finding a shared sentiment subspace

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

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

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



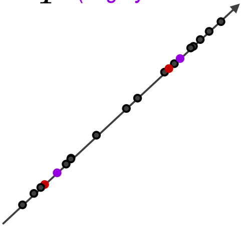
Finding a shared sentiment subspace

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

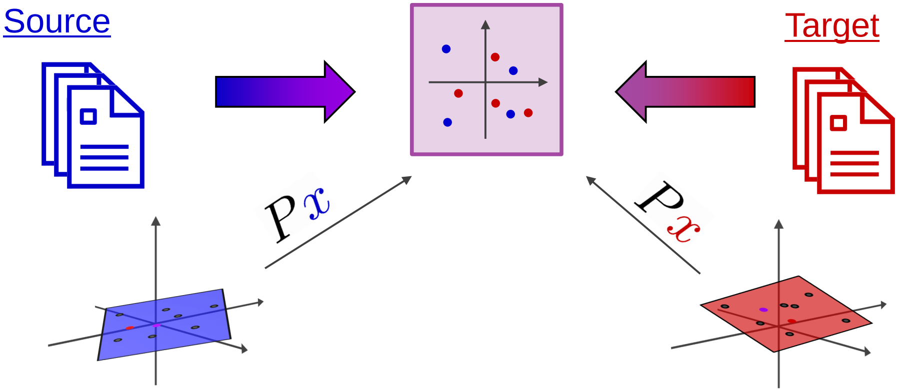
- Let P be a basis for the subspace of best fit to W
- P captures sentiment variance in W

- $p_W(\textit{pivots}|x)$ 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*)

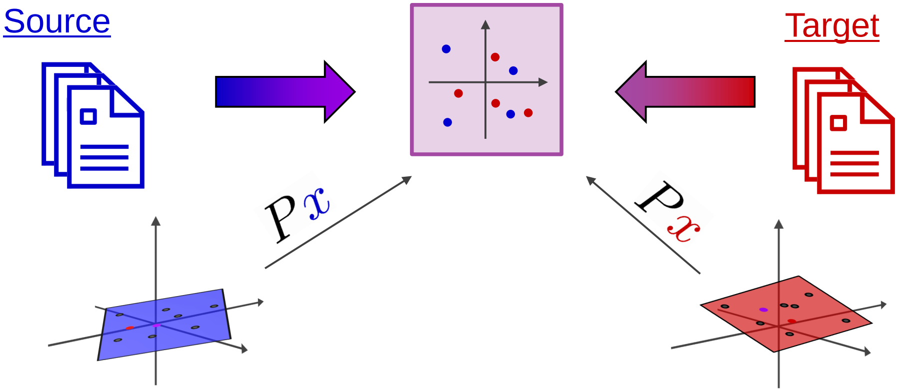


P projects onto shared subspace



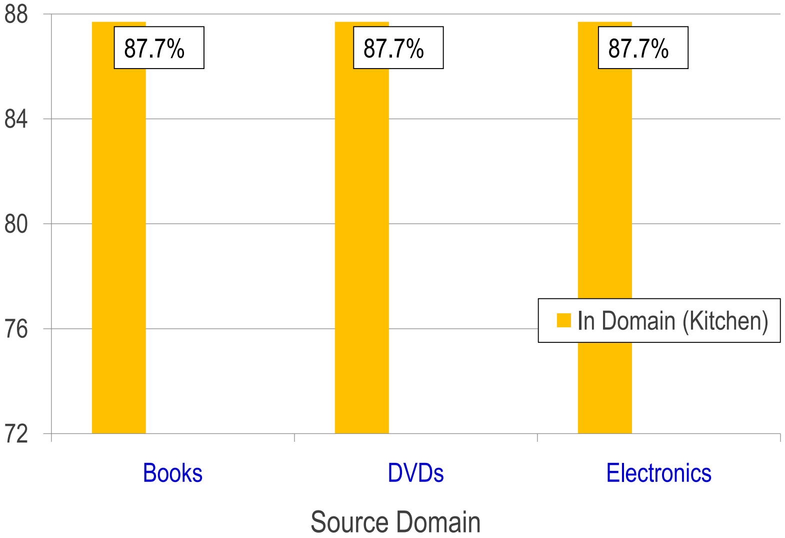
$$p_{\tilde{\theta}}(\text{👍}|x) \propto \exp\left\{\langle \phi(\text{👍}, Px), \tilde{\theta}\rangle\right\}$$

P projects onto shared subspace

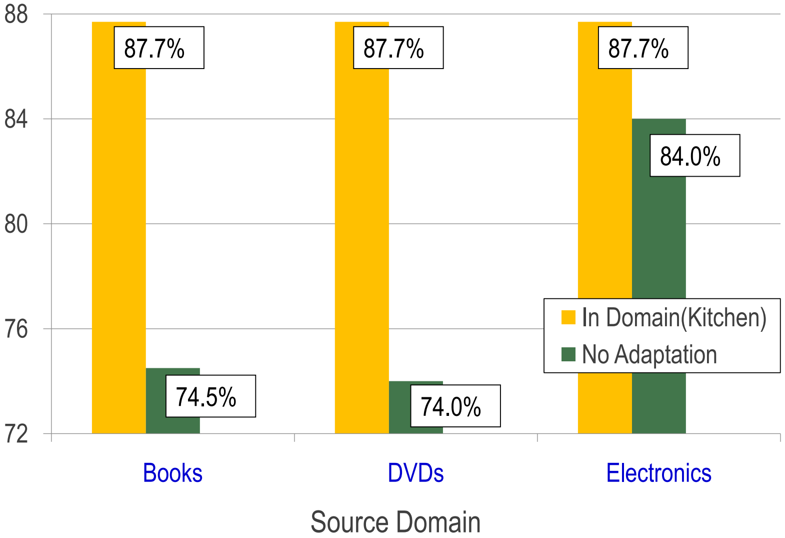


$$h(x) = \operatorname{sgn}\left(\theta^\top P x\right)$$

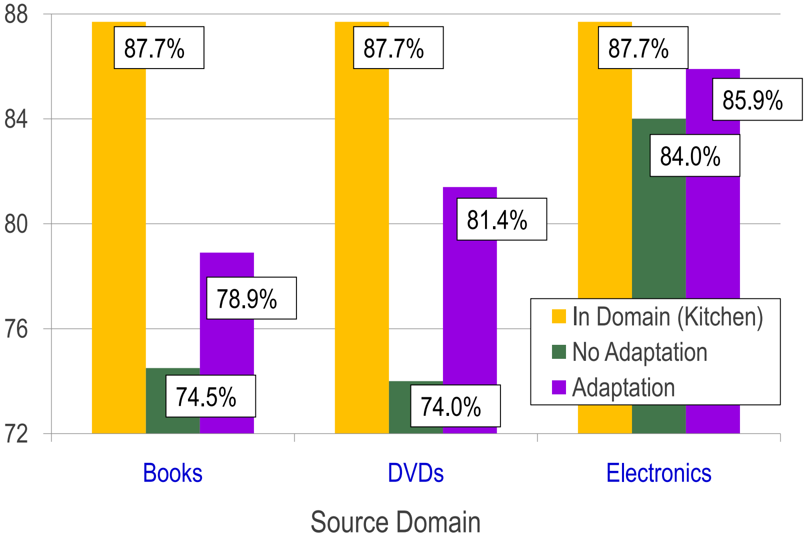
Target Accuracy: Kitchen Appliances



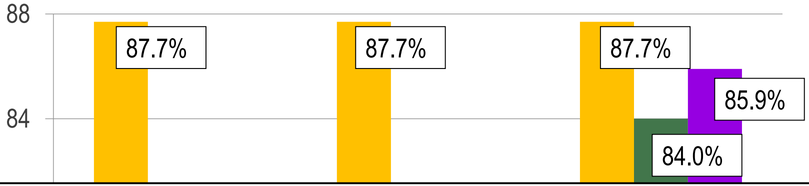
Target Accuracy: Kitchen Appliances



Target Accuracy: Kitchen Appliances



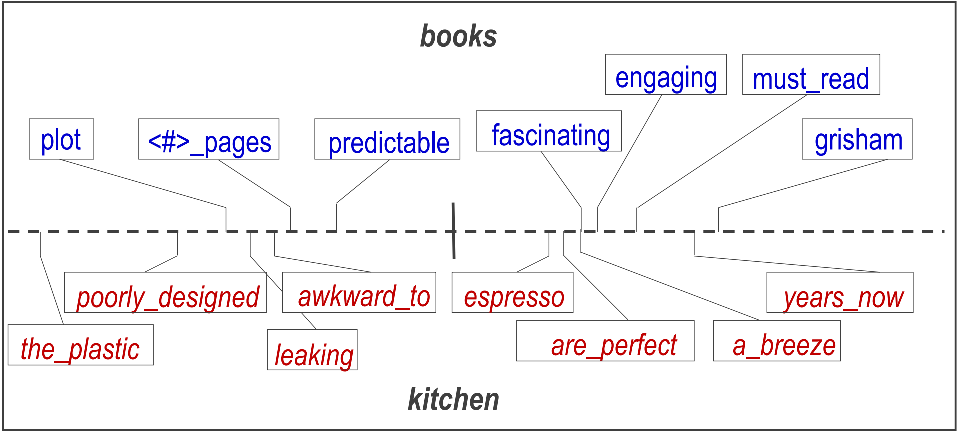
Adaptation Error Reduction



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

Visualizing P (books & kitchen)

negative vs. positive



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)