Reading the Web:
Advanced Statistical Language Processing

www.cs.cmu.edu/~tom/rtw09/

Machine Learning 10-709

October 8, 2009

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

When can Unlabeled Data help supervised learning?

Problem setting (the PAC learning setting):

- Set *X* of instances drawn from unknown distribution *P(X)*
- Wish to learn target function *f: X→ Y* (or, P(Y|X))
- Given a set *H* of possible hypotheses for *f*

Given:

- i.i.d. labeled examples $L = \{\langle x_1, y_1\rangle \ldots \langle x_m, y_m\rangle\}$
- i.i.d. unlabeled examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Wish to find hypothesis with lowest true error:

$$\hat{f} \leftarrow \arg\min_{h \in H} \Pr_{x \in P(X)}[h(x) \neq f(x)]$$

What if x_i are fully observed?

Can use $U \rightarrow \hat{P}(X)$ to alter optimization problem

- Wish to find

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) P(x)$$

- Often approximate as

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \frac{1}{|L|} \sum_{\langle x, y \rangle \in L} \delta(h(x) \neq y)$$

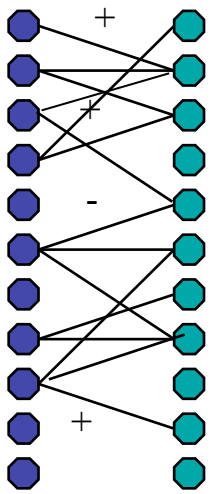
  and when $y = f(x)$, this is just

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x, L)}{|L|}$$

- Can use U for improved approximation:

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x, L) + n(x, U)}{|L| + |U|}$$

What if CoTraining Assumption Not Perfectly Satisfied?



- Idea: Want classifiers that produce a *maximally consistent* labeling of the data
- If learning is an optimization problem, what function should we optimize?

Co-EM [Nigam & Ghani, 2000; Jones 2005]

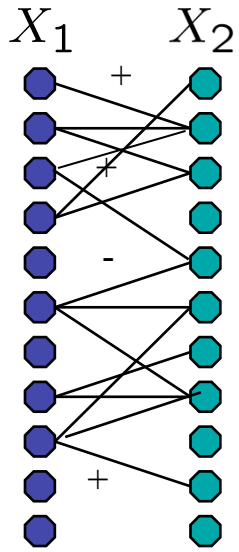
Idea:

- Like co-training, train two coupled functions
  - P(class | X1), P(class | X2)
- Like EM, iterative probabilistic algorithm
  - Assign probabilistic values to unobserved class labels
  - Updating model parameters (= labels of other feature set)

Goal to learn $X_1 \rightarrow Y,\ X_2 \rightarrow Y,\ X_1 \times X_2 \rightarrow Y$

$$P(Y|X_1 = k) = \sum_j P(Y|X_2 = j)P(X_2 = j|X_1 = k)$$

$$P(Y|X_2 = j) = \sum_k P(Y|X_1 = k)P(X_1 = k|X_2 = j)$$



CoRegularization

Key idea:

- define explicit learning objective
- optimize it directly

What objective?

What Objective Function?

$$E = E1 + E2$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

Error on labeled examples

What Objective Function?

$$E = E1 + E2 + c_3 E3$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Error on labeled examples

Disagreement over unlabeled

What Objective Function?

$$E = E1 + E2 + c_3 E3 + c_4 E4$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

Error on labeled examples

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

Disagreement over unlabeled

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Misfit to estimated class priors

$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y>\in L} y \right) - \left(\frac{1}{|L|+|U|} \sum_{x\in L\cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

What Function Approximators?

What Function Approximators?

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}}$$

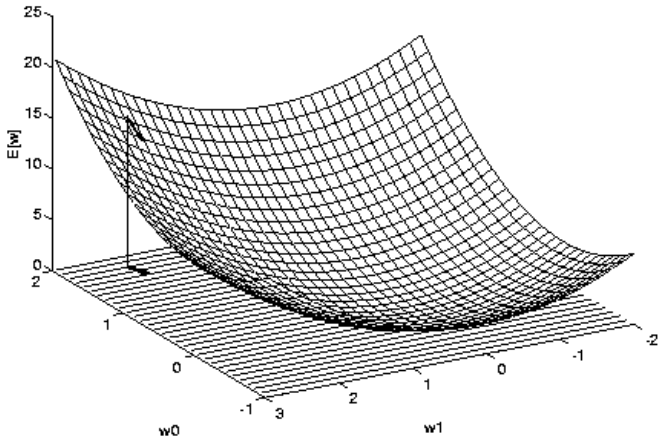
$$\hat{g}_2(x) = \frac{1}{1 + e^{\sum_k w_{k,2} x_k}}$$

- Same fn form as Logistic regression, Max Entropy

- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4

Gradient CoTraining

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}} \qquad \hat{g}_2(x) = \frac{1}{1 + e^{\sum_j w_{j,2} x_j}}$$



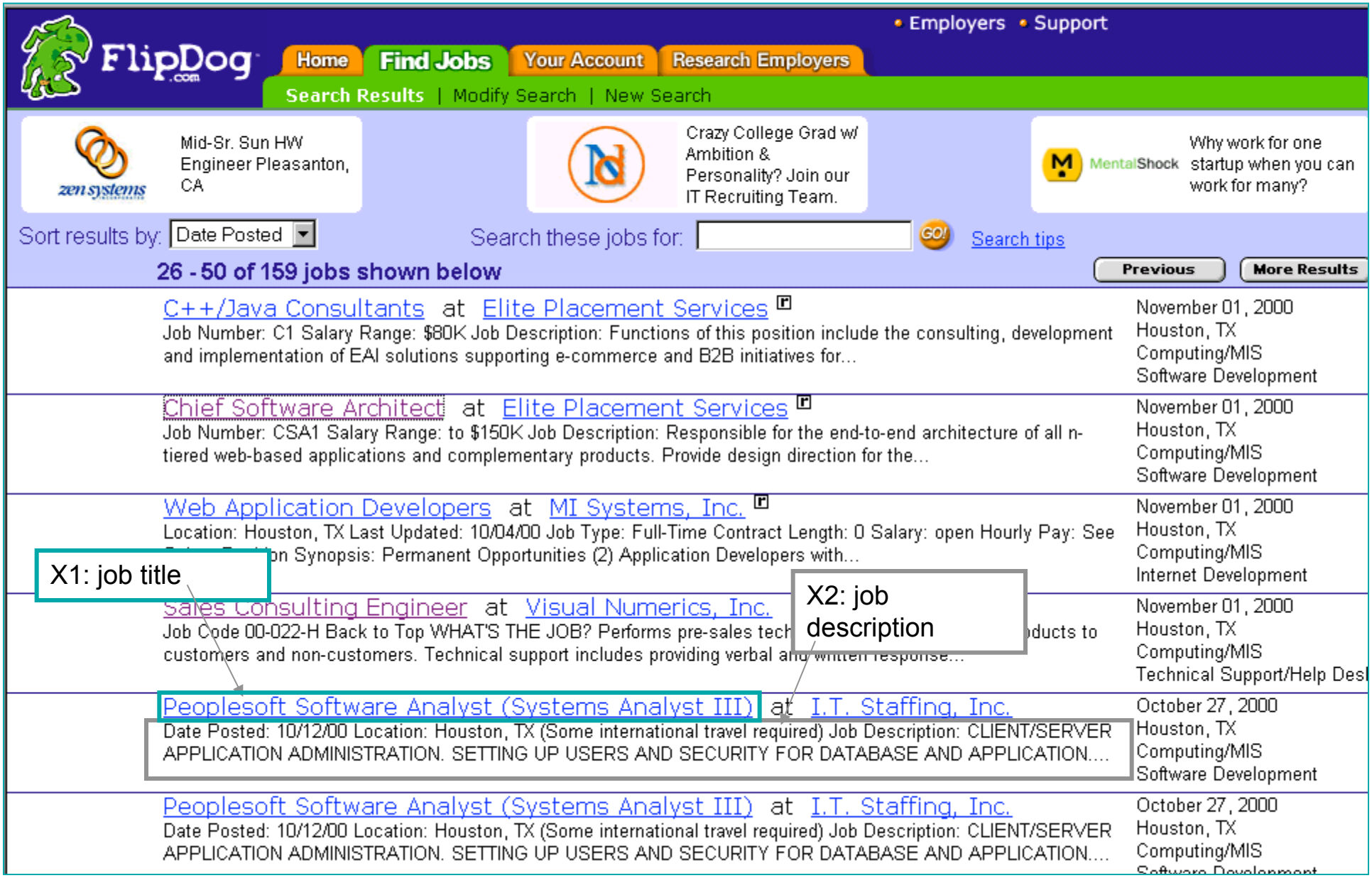
Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

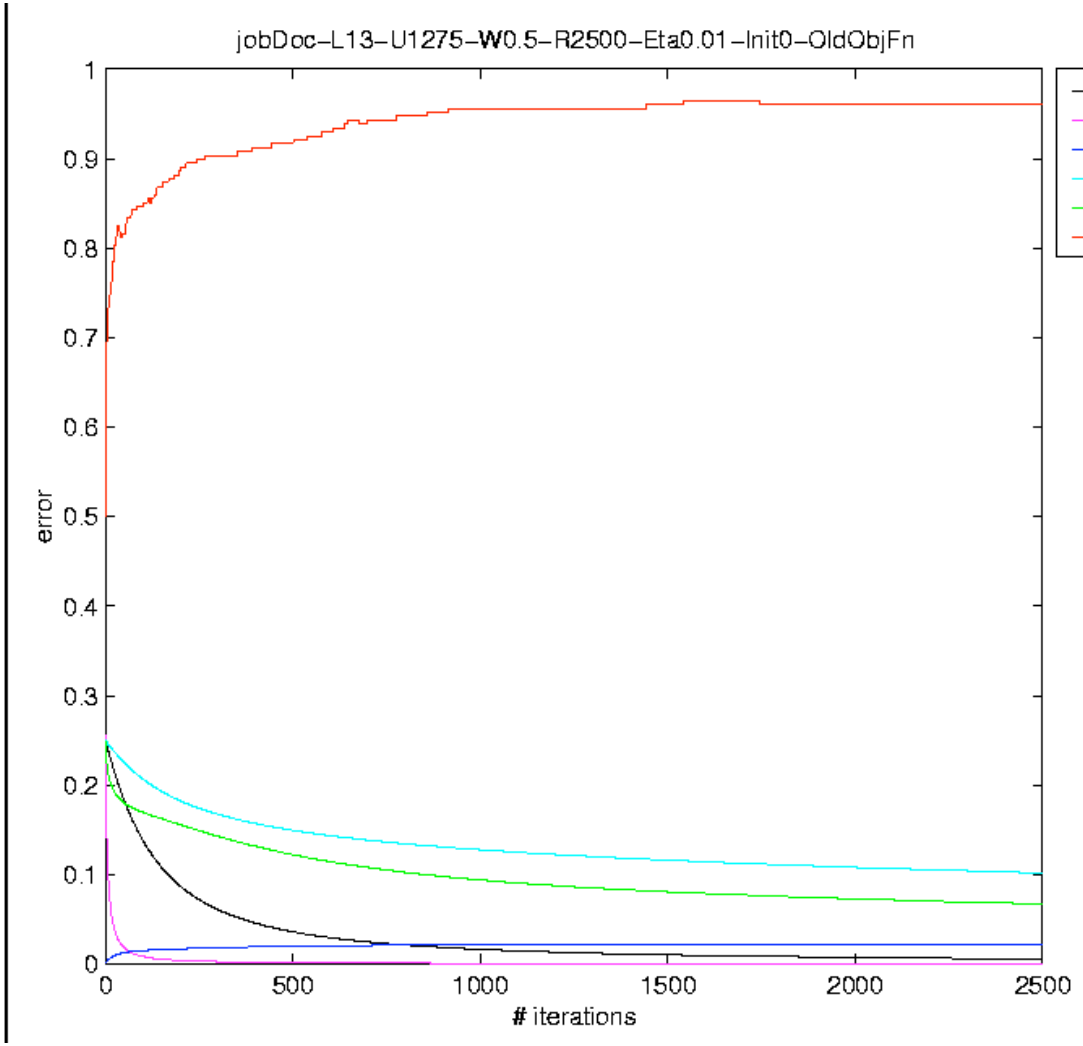
$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

Classifying Jobs for FlipDog



CoRegularization

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoRegularization: 96%

CoRegularization

Classifying Upper Case sequences as Person Names

Error Rates

	25 labeled 5000 unlabeled	***2300 labeled 5000 unlabeled***
Using labeled data only	.24	.13
CoRegularization	.15 *	.11 *
CoRegularization without fitting class priors (E4)	.27 *	

* sensitive to weights of error terms E3 and E4

CoTraining/CoRegularization

- Unlabeled data improves supervised learning when example features are redundantly sufficient
  - Family of algorithms that train multiple classifiers

- Theoretical results
  - Expected error for rote learning
  - If X1,X2 conditionally independent given Y, Then
    - PAC learnable from weak initial classifier plus unlabeled data
    - disagreement between g1(x1) and g2(x2) bounds final classifier error

- Many real-world problems of this type
  - Semantic lexicon generation [Riloff, Jones 99], [Collins, Singer 99]
  - Web page classification [Blum, Mitchell 98]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]
  - Visual classification of cars [Levin, Viola, Freund 03]

Coupled training type 2

Wish to learn f1: X → Y1, f2: X → Y2,

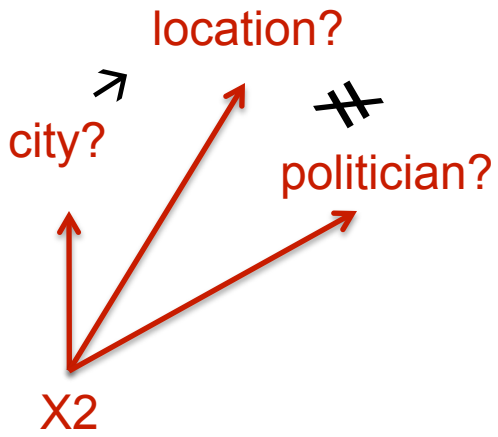
such that: (∀ x) g(f1(x), f2(x))

e.g.

location: NounPhraseInSentence → {0,1}

politician: NounPhraseInSentence → {0,1}

g(y1,y2) = not (and(y1,y2))



Luke is mayor of Pittsburgh.

Coupling functions with different outputs

[Daume, 2008]

Wish to learn f1: X → Y1, f2: X → Y2,

such that: (∀x) g(f1(x), f2(x))

Key theoretical question: what is sample complexity? How does it depend on the coupling constraint, g?

Key insight:

- g will be most useful if the probability that it is satisfied by a high error f applied to a random x, is low

Coupling functions with different outputs

[Daume, 2008]

Consider simpler one-sided learning of f2, given we know f1

1: Learn h_2 directly on D
2: For each example $(x, y_1) \in D^{\text{unlab}}$
3: Compute $y_2 = h_2(x)$
4: If $\chi(y_1, y_2)$, add (x, y_2) to D
5: Relearn h_2 on the (augmented) D
6: Go to (2) if desired

Definition 4. *We say the* discrimination *of* χ *for* h^0 *is* $\Pr_{\mathcal{D}}[\chi(f_1(x), h^0(x))]^{-1}$.

Coupling functions with different outputs

[Daume, 2008]

Theorem 1. *Suppose C_2 is PAC-learnable with noise in the structured setting, h_2^0 is a weakly useful predictor of f_2, and χ is correct with respect to $\mathcal{D}, f_1, f_2, h_2^0$, and has discrimination $\geq 2(|\mathcal{Y}| - 1)$. Then C_2 is also PAC-learnable with one-sided hints.*

(here |Y|= |Y1| x |Y2| is the number of values the two functions can take on)

Structured Output Learning

Suppose we wish to learn f: X → Y

where Y is a vector, tree, or graph?

Want to learn simultaneously the dependencies among components of Y, and their dependence on X

Conditional Random Fields

see Sutton & McCallum, “An Introduction to Conditional Random Fields for Relational Learning”

Conditional Random Fields

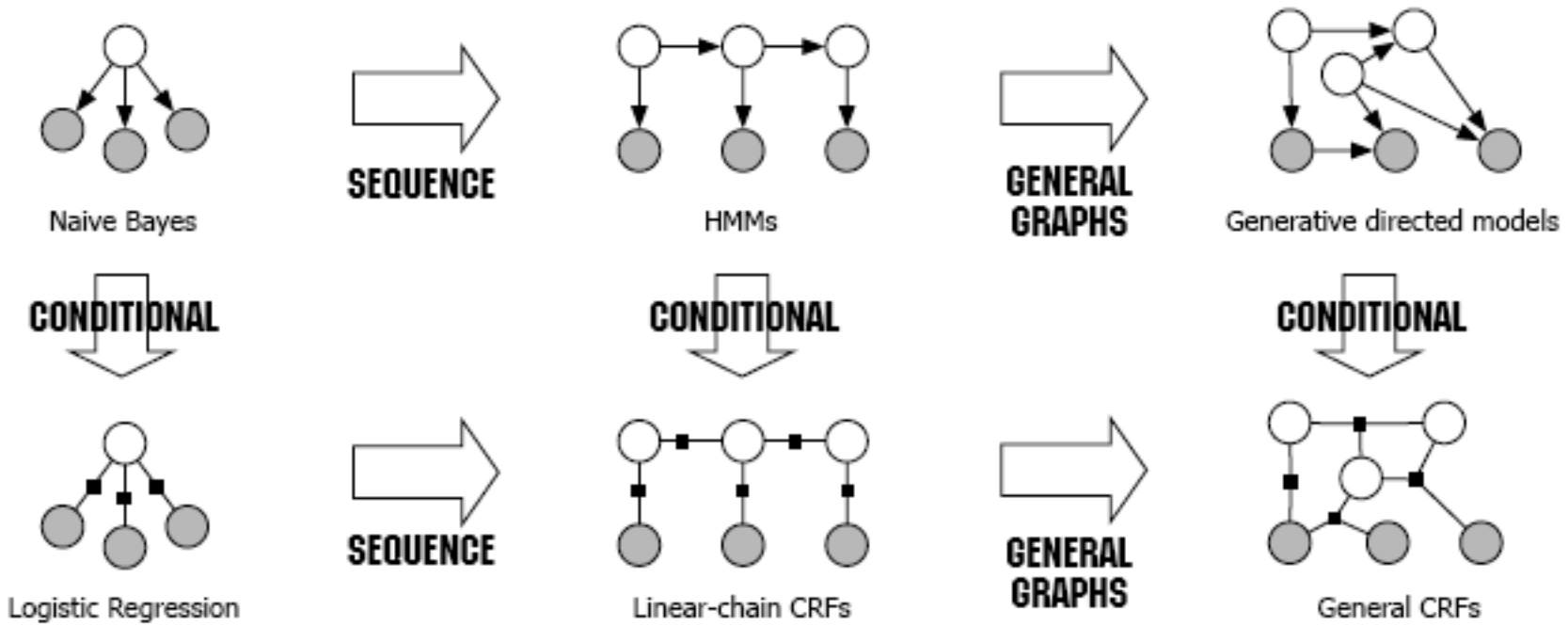


Figure 1.2 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

Factor Graphs

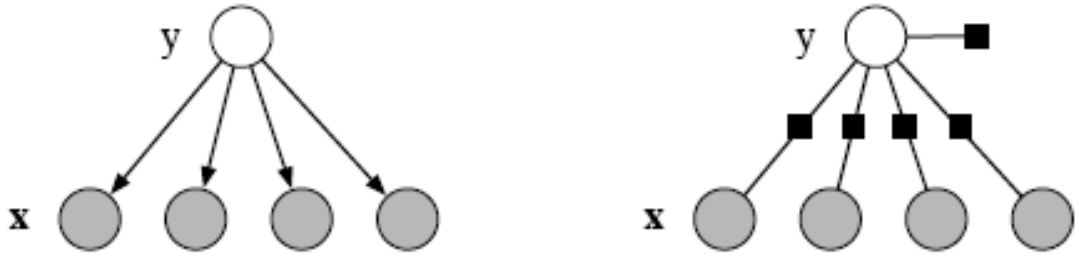


Figure 1.1 The naive Bayes classifier, as a directed model (left), and as a factor graph (right).

an *undirected graphical model* as the set of all distributions that can be written in the form

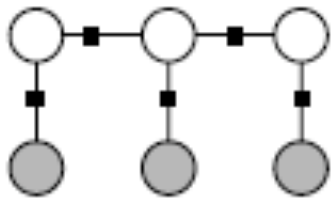
$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_A \Psi_A(\mathbf{x}_A, \mathbf{y}_A), \tag{1.1}$$

for any choice of *factors* $F = \{\Psi_A\}$, where $\Psi_A : \mathcal{V}^n \to \Re^+$. (These functions are also called *local functions* or *compatibility functions*.) We will occasionally use the term *random field* to refer to a particular distribution among those defined by an undirected model. To reiterate, we will consistently use the term *model* to refer to a family of distributions, and *random field* (or more commonly, distribution) to refer to a single one.

The constant Z is a normalization factor defined as

$$Z = \sum_{\mathbf{x},\mathbf{y}} \prod_A \Psi_A(\mathbf{x}_A, \mathbf{y}_A), \tag{1.2}$$

Linear Chain CRF



Linear-chain CRFs

Definition 1.1

Let Y, X be random vectors, $\Lambda = \{\lambda_k\} \in \Re^K$ be a parameter vector, and $\{f_k(y, y', \mathbf{x}_t)\}_{k=1}^K$ be a set of real-valued feature functions. Then a *linear-chain conditional random field* is a distribution $p(\mathbf{y}|\mathbf{x})$ that takes the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}, \tag{1.16}$$

where $Z(\mathbf{x})$ is an instance-specific normalization function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}. \tag{1.17}$$

We have just seen that if the joint $p(\mathbf{y}, \mathbf{x})$ factorizes as an HMM, then the associated conditional distribution $p(\mathbf{y}|\mathbf{x})$ is a linear-chain CRF. This HMM-like CRF is

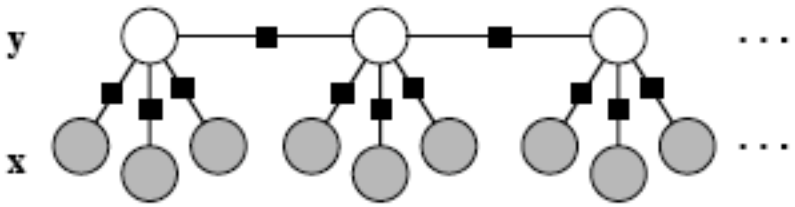


Figure 1.3 Graphical model of an HMM-like linear-chain CRF.

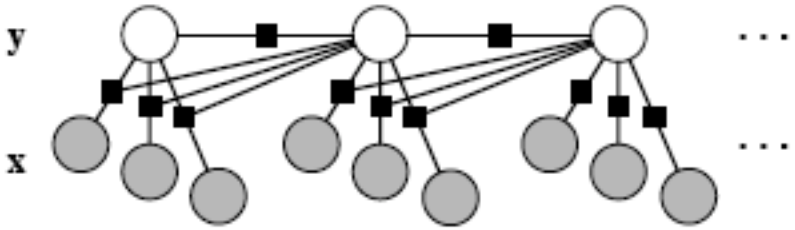


Figure 1.4 Graphical model of a linear-chain CRF in which the transition score depends on the current observation.

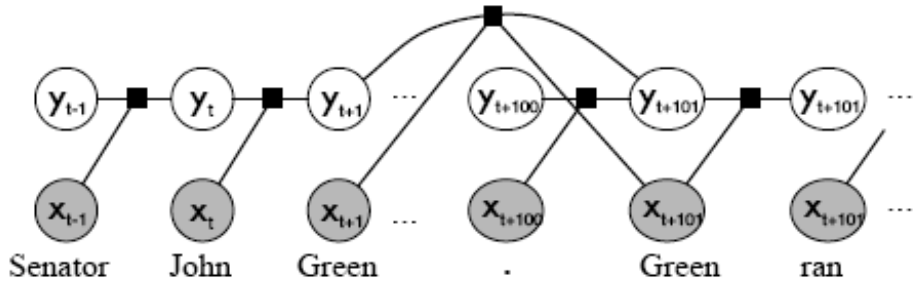


Figure 1.5 Graphical representation of a skip-chain CRF. Identical words are connected because they are likely to have the same label.

"skip" links added only for identical, capitalized tokens

$w_t = w$
w_t matches `[A-Z][a-z]+`
w_t matches `[A-Z][A-Z]+`
w_t matches `[A-Z]`
w_t matches `[A-Z]+`
w_t matches `[A-Z]+[a-z]+[A-Z]+[a-z]`
w_t appears in list of first names, last names, honorifics, etc.
w_t appears to be part of a time followed by a dash
w_t appears to be part of a time preceded by a dash
w_t appears to be part of a date
$T_t = T$
$q_k(\mathbf{x}, t+\delta)$ for all k and $\delta \in [-4, 4]$

task: label seminar speaker, location, start time, end time

Table 1.1 Input features $q_k(\mathbf{x}, t)$ for the seminars data. In the above w_t is the word at position t, T_t is the POS tag at position t, w ranges over all words in the training data, and T ranges over all part-of-speech tags returned by the Brill tagger. The "appears to be" features are based on hand-designed regular expressions that can span several tokens.

System	stime	etime	location	speaker	overall
BIEN Peshkin and Pfeffer [2003]	96.0	**98.8**	87.1	76.9	89.7
Linear-chain CRF	**97.5**	97.5	**88.3**	77.3	90.2
Skip-chain CRF	96.7	97.2	88.1	**80.4**	**90.6**

Table 1.2 Comparison of F_1 performance on the seminars data. The top line gives a dynamic Bayes net that has been previously used on this data set. The skip-chain CRF beats the previous systems in overall F1 and on the speaker field, which has proved to be the hardest field of the four. Overall F1 is the average of the F1 scores for the four fields.

Homework 3

- http://www.cs.cmu.edu/~tom/10709_fall09/hw3.html

Further Reading

- Semi-Supervised Learning, O. Chapelle, B. Sholkopf, and A. Zien (eds.), MIT Press, 2006. (excellent book)

- Semi-Supervised Learning for Computational Linguistics, S. Abney, Springer, 2007. (pretty good, pretty basic)

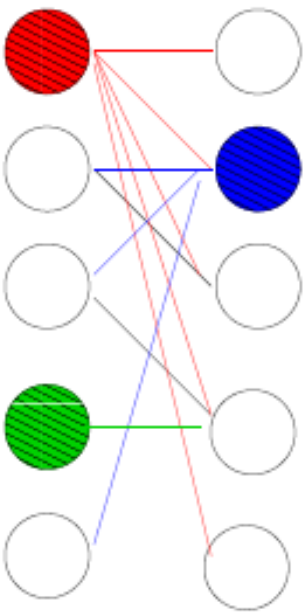
- EM for Naïve Bayes classifiers: K.Nigam, et al., 2000. "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,* 39, pp.103—134.

- CoTraining: A. Blum and T. Mitchell, 1998. “Combining Labeled and Unlabeled Data with Co-Training,” *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98).*

- S. Dasgupta, et al., “PAC Generalization Bounds for Co-training”, *NIPS 2001*

- Model selection: D. Schuurmans and F. Southey, 2002. “Metric-Based methods for Adaptive Model Selection and Regularization,” Machine Learning, 48, 51—84.

Some nodes are more important than others [Jones, 2005]



Can use this for active learning...

Noun-phrase	Outdegree
you	1656
we	1479
it	1173
company	1043
this	635
all	520
they	500
information	448
us	367
any	339
products	332
i	319
site	314
one	311
1996	282
he	269
customers	269
these	263
them	263
time	234

Context	Outdegree
<x> including	683
including <x>	612
<x> provides	565
provides <x>	565
provide <x>	390
<x> include	389
include <x>	375
<x> provide	364
one of <x>	354
<x> made	345
<x> offers	338
offers <x>	320
<x> said	287
<x> used	283
includes <x>	279
to provide <x>	266
use <x>	263
like <x>	260
variety of <x>	252
<x> includes	250