CSCI 5622 Machine Learning

ML Semi-Supervised Learning

DATE	READ	DUE
Today, Nov 4	***Semi-supervised Learning***	
Mon, Nov 9	SSL continued	Exprmnt 1 Write-up
Wed, Nov 11	Active Learning	

www.RodneyNielsen.com/teaching/CSCI5622-F09/

Instructor: Rodney Nielsen

Assistant Professor Adjunct, CU Dept. of Computer Science

Research Assistant Professor, DU, Dept. of Electrical & Computer Engr.

Research Scientist, Boulder Language Technologies

Evaluating Clustering Results

ML

- Goals:
  - High intra-cluster similarity (cluster purity)
  - Low inter-cluster similarity (cluster uniqueness)
- However, the clustering quality is dependent
  - Not only on the metrics
  - But also the application
- Therefore, evaluation in context is better
  - But is rarely done

Clustering Metrics

ML

- Purity
  - Measures the level of intra-cluster similarity
- Normalized Mutual Information
  - Provides an information theoretic measure
- Rand Index
  - Combines both intra- and inter-cluster assessment
- *F*-measure
  - Also factors in both intra- and inter-cluster quality

ML Questions

- Questions???

Partially Observable Data

- Some feature values occasionally missing
  - Estimate missing values
    - Most common value (for that class)
    - Mean value (for that class)
    - Learn the value (for that class)
    - In DTs, assume fractional values based on proportions
- When values, θ, are never observed?
  - EM Algorithm can learn these values, provided that their probability distribution is known

EM Algorithm

- Generate an arbitrary hypothesis *h*
  - I.E., make arbitrary assignments to $\boldsymbol{\theta}$
- The (**Expectation**) **E**-Step: Based on the current hypothesis *h*, calculate the ***Expected*** values of the class indicator variables
  $\mathbf{Y} = \{\mathbf{y}^{(i)}\} = \{\langle y_1^{(i)}, y_2^{(i)}, \ldots, y_K^{(i)}\rangle\}$
- The (**Maximization**) **M**-Step: Based on the expected value of the data, $\{\langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)}\rangle\}$, compute *h* that ***Maximizes*** the likelihood of the data
  $h_{ML} = \boldsymbol{\theta}_{ML}$

ML Ex: One Attribute & Two Classes

- (Expectation) E-Step:

$$E\left[y_1^{(i)}\right] = \frac{p\left(x = x^{(i)} \middle| \mu = \mu_1\right)}{\sum_{k=1}^{2} p\left(x = x^{(i)} \middle| \mu = \mu_k\right)}$$

$$= \frac{\exp\left(-\frac{1}{2\sigma^2}\left(x^{(i)} - \mu_1\right)^2\right)}{\exp\left(-\frac{1}{2\sigma^2}\left(x^{(i)} - \mu_1\right)^2\right) + \exp\left(-\frac{1}{2\sigma^2}\left(x^{(i)} - \mu_2\right)^2\right)}$$

ML Ex: One Attribute & Two Classes

- (Maximization) M-Step:

$$\mu_{1,ML} = \frac{\sum_{i=1}^{N} E\left[y_1^{(i)}\right] x^{(i)}}{\sum_{i=1}^{N} E\left[y_1^{(i)}\right]}$$

ML Semi-Supervised Learning (SSL)

- Supervised ML often requires a significant amount of data to achieve reasonable results
  - SRL: 1M words text → ~ 2M training examples
    - ~80% F-measure on out-of-domain data
- But Google indexes trillions of words
- Can we use all of that unlabeled data to improve ML performance

ML Applications

- Virtually all NLP tasks and applications:
  - POS tagging, Parsing, Semantic Role Labeling, Entity and event detection, …
  - Information Extraction, Information Retrieval, Document Classification, Educational Assessment,…
- Virtually all image processing tasks & apps:
  - Edge detection, shape & texture recognition,…
  - Face recognition, object recognition,…
- Vertical profiling: waveform identification,…

ML Document Classification

Text Classification from Labeled and Unlabeled Documents using EM

KAMAL NIGAM[†] knigam@cs.cmu.edu
ANDREW KACHITES MCCALLUM[‡†] mccallum@justresearch.com
SEBASTIAN THRUN[†] thrun@cs.cmu.edu
TOM MITCHELL[†] tom.mitchell@cmu.edu
[†] *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213*
[‡] *Just Research, 4616 Henry Street, Pittsburgh, PA 15213*

Received March 15, 1998; Revised February 20, 1999
Editor: William W. Cohen

Abstract. This paper shows that the accuracy of learned text classifiers can be improved by augmenting a small number of labeled training documents with a large pool of unlabeled documents. This is important because in many text classification problems obtaining training labels is expensive, while large quantities of unlabeled documents are readily available.

We introduce an algorithm for learning from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and a naive Bayes classifier. The algorithm first trains a classifier using the available labeled documents, and probabilistically labels the unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates to convergence. This basic EM procedure works well when the data conform to the generative assumptions of the model. However these assumptions are often violated in practice, and poor performance can result. We present two extensions to the algorithm that improve classification accuracy under these conditions: (1) a weighting factor to modulate the contribution of the unlabeled data, and (2) the use of multiple mixture components per class. Experimental results, obtained using text from three different real-world tasks, show that the use of unlabeled data reduces classification error by up to 30%.

Keywords: text classification, Expectation-Maximization, integrating supervised and unsupervised learning, combining labeled and unlabeled data, Bayesian learning

1. Introduction

Consider the problem of automatically classifying text documents. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Internet news feeds, electronic mail, corporate databases, medical patient records and digital libraries. Existing statistical text learning algorithms can be trained to approximately classify documents, given a sufficient set of labeled training examples. These text classification algorithms have been used to automatically catalog news articles (Lewis & Gale, 1994; Joachims, 1998) and web pages (Craven, DiPasquo, Freitag, McCallum, Mitchell, Nigam, & Slattery, 1998; Shavlik & Eliassi-Rad, 1998), automatically learn the reading interests of users (Pazzani, Muramatsu, & Billsus, 1996; Lang, 1995), and automati-

→

0	aardvark
1	abstract
4	computer
0	dumpling
178	labeled
92	learning
17	machine
7	science
212	unlabeled
0	zephyr

Naïve Bayes

- Fully supervised approach

$$\hat{y} = \arg\max_k P(c_k | doc_n)$$

$$= \arg\max_k \frac{P(c_k) P(doc_n | c_k)}{P(doc_n)}$$

$$= \arg\max_k P(c_k) P(doc_n | c_k)$$

$$= \arg\max_k P(c_k) \prod_{w_i \in doc_n} P(w_i | c_k)$$

$$= \arg\max_k \frac{\mathrm{N}(c_k) + 1}{N + K} \prod_{w_i \in doc_n} \frac{\mathrm{N}(w_i, c_k) + 1}{\mathrm{N}(c_k) + |V|}$$

ML If we don't have all the labels?

	CLS LBL	dumpling	labeled	learning	machine	science	zephyr
Doc_1	1	0	178	92	17	7	0
Doc_2	0	15	1	1	0	0	1
Doc_3	1	0	28	32	10	3	1
Doc_4	?	0	0	7	5	5	0
Doc_5	?	0	0	5	0	0	5

- Use Semi-Supervised Learning
- One option is EM

Apply EM to Naïve Bayes

- Initialize
  - $P(y=c_k)$ or $P(c_k)$ and $P(w_v|c_k)$
  - based on the labeled data
- E-Step
  - Compute ***expected*** value for doc. class labels
- M-Step
  - ***Maximize*** the likelihood of data given the labels

ML Ex: One Attribute & Two Classes

- E-Step (Expectation):

$$
\begin{aligned}
E\left[y_k^{(n)}\right] &= P\left(c_k \middle| doc_n;\theta\right) \\
&= \frac{P\left(c_k \middle|\theta\right) P\left(doc_n \middle| c_k;\theta\right)}{P\left(doc_n \middle|\theta\right)} \\
&= \frac{P\left(c_k;\theta\right) \prod_{w_i \in doc_n} P\left(w_i \middle| c_k;\theta\right)}{\sum_{h=1}^{K}\left(P\left(c_h;\theta\right) \prod_{w_i \in doc_n} P\left(w_i \middle| c_h;\theta\right)\right)}
\end{aligned}
$$

ML Ex: One Attribute & Two Classes

- (Maximization) M-Step:
  - Recompute **θ**

$$P\left(c_k \middle| \theta_{t-1}\right) = \frac{1 + \sum_{i=1}^{N} P\left(c_k \middle| doc_n; \theta_{t-1}\right)}{N + K}$$

$$P\left(w_l \middle| c_k; \theta_{t-1}\right) = \frac{1 + \sum_{n=1}^{N} \mathrm{N}\left(w_l, doc_n\right) P\left(c_k \middle| doc_n; \theta_{t-1}\right)}{\left|V\right| + \sum_{s=1}^{|V|} \sum_{n=1}^{N} \mathrm{N}\left(w_s, doc_n\right) P\left(c_k \middle| doc_n; \theta_{t-1}\right)}$$

ML Modifications

- Reduce the weight of the unlabeled examples
- Could be extended to other distributions or models (ie, Bayes Net)

When will this work?

ML

- If and only if the documents in a given class are relatively consistent as if generated by a single process

ML Questions

- Questions???

Co-Training

ML

- Allow two classifiers given independent views to learn from one another
- Each attribute set (view) must be sufficient
- Each view must be independent of the other given the class labels

Ex: Document Classification

- Classify computer science department web pages as course home pages
- Two views:
  - The words on the page to be classified
  - The displayed text of the hyperlinks that connect to the page to be classified

Co-Training Algorithm 1

- Given
  - Labeled data *L*
  - Unlabeled data *U*
- Loop
  - Train h_1 (hyperlink classifier) using instances in *L*
  - Train h_2 (page classifier) using instances in *L*
  - Have h_1 and h_2 each classify instances in U
  - Have h_1 and h_2 each add *p* positively- and *n* negatively-labeled instances from *U* to *L*

Ex: Document Classification

- Classify course home pages
- Two views: pages versus hyperlinks
- Started with just 3 positive examples and 9 negative examples
- 1000 additional unlabeled examples
- Achieved an error rate of 5%
- Reduced the error rate by over 50% over just using the labeled data

Ex: Document Classification

From Blum and Mitchell (1998)

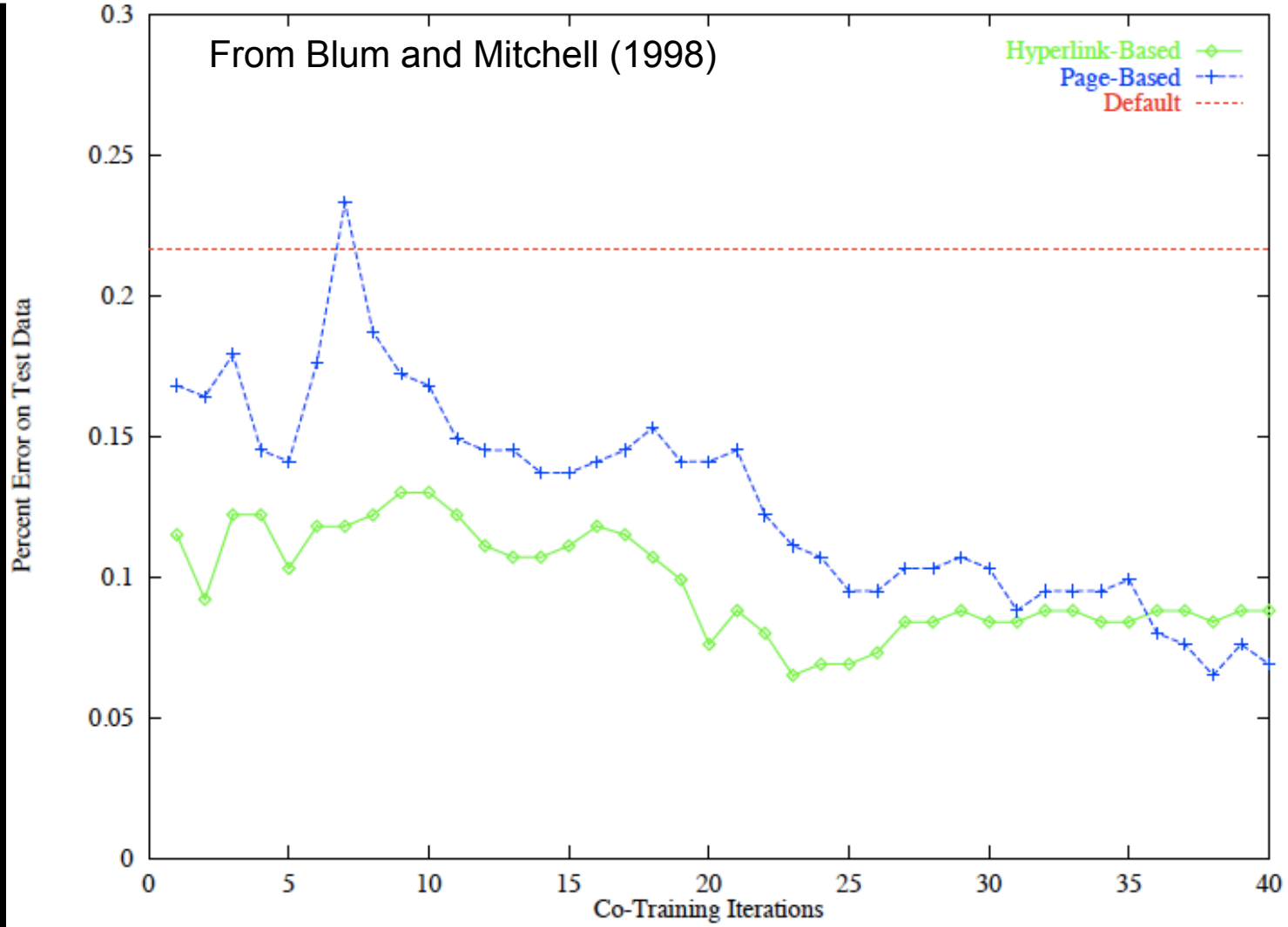

Figure 2: Error versus number of iterations for one run of co-training experiment.

Goals

- Both classifiers should
  - Correctly classify the labeled examples
    - *f* must be PAC learnable by each classifier
  - Agree on the classification of the unlabeled data, eventually

Theorem

- If
  - **X** and **X'** (the two views) are conditionally independent given the labels, and
  - f is PAC learnable using either view
- Then
  - f is PAC learnable from weak initial classifiers plus unlabeled data

Co-EM

ML

- Like Co-training: 2 separate views of the data
- Like EM: Iteratively update unobserved values
  - Use one classifier to update the other

Yarowski Algorithm

ML

- Word Sense Disambiguation
- View 1: the other words within the context window of the word to be disambiguated
- View 2: the document
  - Assume one sense per document

ML Questions

- Questions???

Bootstrapping NE Tagging

ML

- Named Entity (NE) tagging involves detecting references to people, locations, times, etc.
- Bootstrapping here is similar to Co-training
- Example:
  - View 1: Reference text of names (e.g., a gazetteer)
  - View 2: Context of NEs in text to be tagged

    Denver The conference was in *Denver*

    Seattle *Seattle*, Washington

ML Questions

- Questions???

Issues in Experimental Design

ML

- Questions about issues in experimental design?

Presentations

ML

- Mon, Nov 9: Susan Philipose
- Wed, Nov 11: Chenyu Zheng
- Volunteers
  - Mon, Nov 16:
  - Wed, Nov 18:

Projects

ML

- Office hours: will stay until all questions are addressed
- Or make an appointment