MTTTS16 Learning from Multiple Sources

5 ECTS credits

Autumn 2019, Tampere University

Lecturer: Jaakko Peltonen

Lecture 7: Multi-view learning for classification by **Co-training**

On this lecture:

- co-training for combining labeled+unlabeled data in predictive tasks
- analysis of co-training
- co-EM: an alternative algorithm somewhat related to expectation maximization
- Bayesian co-training

Part 1:

Co-training - Multi-view learning to combine labeled and unlabeled data

Co-training: multiple views, labeled+unlabeled data

- In many predictive tasks, it is hard to get a lot of labeled training data, but unlabeled data may be much easier to get.
- Example 1: to recognize faces in images, it is hard to get lots of labeled face images, but it is easy to get unlabeled faces
- Example 2: to categorize web pages, it takes time to categorize pages, but it is easy to crawl more uncategorized pages
- Labeled samples may be hard to get because it takes human effort, or finding out the label costs money, etc.
- Semisupervised learning is a family of approaches that try to learn predictive tasks using both labeled and unlabeled data

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- Unlabeled input data can help learn the input distribution (probability density function of inputs)
- Usually, semisupervised learning approaches are based on an assumption that the shape of the input distribution has something to do with the probabilities of different outputs (classes)
- For example, semisupervised learning approaches may assume that decision boundaries between classes occur at areas of low input density (valleys of the distribution).
    - This is sometimes called a “cluster assumption”, since it means individual classes are strongest inside clusters of the input density.
- However, not everything about the input density may be related to classes.

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- When multiple views (multiple feature sets) are available for the data, the views may help use unlabeled data more effectively
- For example, web pages may be described 1) by the text of the webpage itself, 2) by the **anchor text** of **hyperlinks pointing to the webpage**
- **Co-training** is an approach to use multiple views for combining labeled and unlabeled data

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- Basic idea:
  - First, find **predictors based on each view** (each kind of information). It is enough to find “weak” predictors that are somewhat better than random.
  - Then use these predictors for **bootstrapping** using unlabeled data
  - In the webpage example, suppose the phrase “my advisor” attached to a hyperlink is a good predictor that the linked webpage is a faculty page.
  - Then we could find unlabeled pages linked to as “my advisor”, label them as faculty pages, then retrain a predictor based on the content of those pages too.
  - Then, using the page-content based predictor, we could label even more unlabeled pages, and retrain the link-based predictor; and so on.

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- We call such bootstrapping **co-training**
- It is similar to bootstrapping from incomplete data in an expectation-maximization setting
- Formally: instance space $X = X_1 \times X_2$ where X_1 and X_2 are the two different views (feature sets)
- Each example is given as a pair (x_1, x_2)
- We make a strong assumption: **each view is itself sufficient for correct classification.**
    - Let $\mathcal{D}$ be a distribution over X, and let C_1 and C_2 be concept classes (function classes) defined over X_1 and X_2 respectively.
    - We assume the labels of examples with non-zero probability on $\mathcal{D}$ are consistent with some function $f_1 \in C_1$ and some function $f_2 \in C_2$.
    - That is, for each $x = (x_1, x_2)$ with label l, $f_1(x_1) = f_2(x_2) = \ell$

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

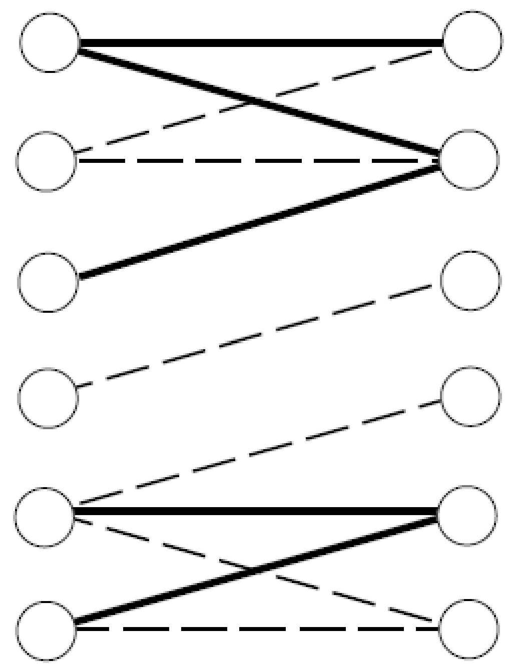
- The assumption implies the distribution $\mathcal{D}$ has zero probability for examples x where $f_1(x_1) \neq f_2(x_2)$
- For a fixed distribution, this implies strong assumptions about the underlying functions f_1 and f_2 : they cannot disagree
- Unlabeled data can then help learn which functions f_1 and f_2 are **compatible** in this sense

Example shown as a graph.
Left side = 1st view,
Right side = 2nd view

Circles = possible values in each view.

Dotted lines = possible samples according to *D*

Solid lines = observed samples



According to the assumption, any values connected through solid lines must get the same class!

“Compatible function classes” = those that partition the graph with no cross-edges

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- One could define a "degree of compatibility" for the two functions and the distribution as the probability of disagreement:

$$p = 1 - \Pr_{\mathcal{D}}[(x_1, x_2) : f_1(x_1) \neq f_2(x_2)]$$

- In reality, views of data may have unavoidable noise (occurring differently in different views) and classifiers with perfect agreement between views may not be possible.
- At the extreme: if all combinations (x_1, x_2) over X_1 and X_2 are possible, then the requirement would mean that every (x_1, x_2) comes from the same class! This is obviously not suitable.
- However, the simple assumption makes it possible to do useful theoretical analysis of whether unlabeled data can help.

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- For example, if we make additional independence assumptions (feature sets are independent given the label):

$$\Pr_{(x_1,x_2)\in\mathcal{D}}\left[x_1 = \hat{x}_1 \mid x_2 = \hat{x}_2\right] = \Pr_{(x_1,x_2)\in\mathcal{D}}\left[x_1 = \hat{x}_1 \mid f_2(x_2) = f_2(\hat{x}_2)\right]$$

$$\Pr_{(x_1,x_2)\in\mathcal{D}}\left[x_2 = \hat{x}_2 \mid x_1 = \hat{x}_1\right] = \Pr_{(x_1,x_2)\in\mathcal{D}}\left[x_2 = \hat{x}_2 \mid f_1(x_1) = f_1(\hat{x}_1)\right]$$

then it turns out a good predictor can be learned from unlabeled data only, as long as we start from at least a **weakly useful initial predictor** (where the conditional probability of the correct class is at least some small value epsilon above the overall predicted proportion of that class in the data)

- The independence assumption turns out to be useful even if we relax the previous assumption of view compatibility.

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- Relaxing the assumptions: instead of assuming perfect view compatibility, in the case of two classes define

$$
\begin{aligned}
p_{11} &= \Pr_{\mathcal{D}}[f_1(x_1)=1, f_2(x_2)=1], \\
p_{10} &= \Pr_{\mathcal{D}}[f_1(x_1)=1, f_2(x_2)=0], \\
p_{01} &= \Pr_{\mathcal{D}}[f_1(x_1)=0, f_2(x_2)=1], \\
p_{00} &= \Pr_{\mathcal{D}}[f_1(x_1)=0, f_2(x_2)=0].
\end{aligned}
$$

- The "perfect compatibility" assumption meant $p_{10} = p_{01} = 0$
- Instead, assume agreement is sufficiently more likely than disagreement: $p_{11}p_{00} \quad > \quad p_{01}p_{10} + \delta$
- Assume we have a weak predictor h from the first view
- Theorem: let $h(x_1)$ be a hypothesis with

$\alpha = \Pr_{\mathcal{D}}[h(x_1)=0 | f_1(x_1)=1]$, $\beta = \Pr_{\mathcal{D}}[h(x_1)=1 | f_1(x_1)=0]$

then $\Pr_{\mathcal{D}}[h(x_1)=0 | f_2(x_2)=1] + \Pr_{\mathcal{D}}[h(x_1)=1 | f_2(x_2)=0]$

$$= 1 - \frac{(1-\alpha-\beta)(p_{11}p_{00} - p_{01}p_{10})}{(p_{11}+p_{01})(p_{10}+p_{00})}.$$

That is, h is also useful for the second view --> co-training can proceed

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

- Co-training algorithm:

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

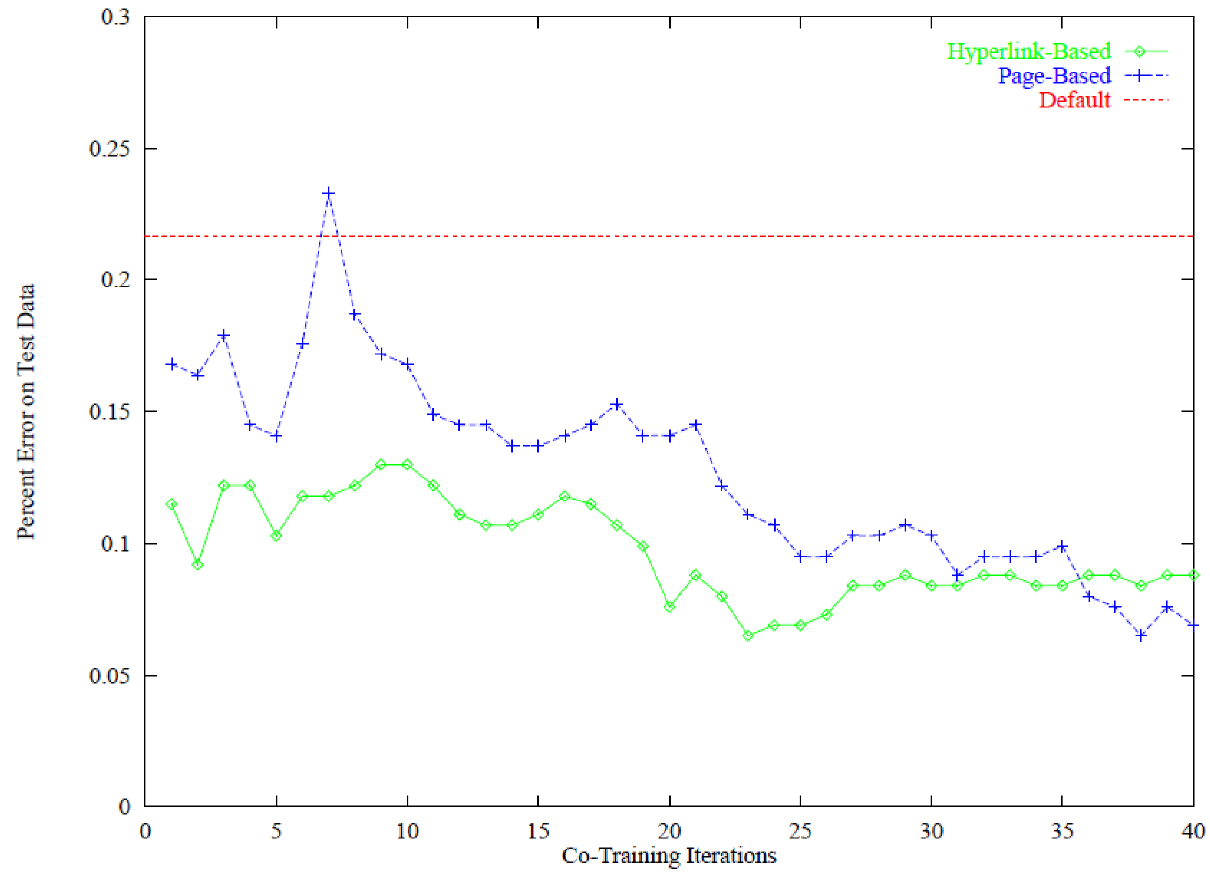
Randomly choose $2p + 2n$ examples from U to replenish U'

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Co-training: multiple views, labeled+unlabeled data

• Experiment: use Naive Bayes classifiers for webpages of 3 universities. Labels: "course homepage" or not. Two views: page-content (bag-of-words vector), and links-to-the-page (bag-of-words vector).

Evolution of cost during training



Final error rates.

Combined classifier: simply multiplies probabilities given by page- and link-based classifiers.

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Following the approach from Blum, Mitchell. Combining labeled and unlabeled data with co-training. In proceedings of COLT 1998. Images from that paper.

Part 2:

More analysis on Co-training
+
the co-EM algorithm

Co-training: questions

• Does co-training make use of independent divisions of features, or does it use unlabeled data only as well as methods that ignore the feature division?

• How sensitive are co-training algorithms to the correctness of their assumptions (compatibility of views, conditional independence of features given the class)?

• Can co-training algorithms be applied to datasets without natural feature divisions?

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: analyzed methods

- Naive Bayes classifier for classes *c* of documents *d* containing words *w*: words occur independently given the class

$$\mathrm{P}(c_j|d_i) \quad \propto \quad \mathrm{P}(c_j)\mathrm{P}(d_i|c_j) \; = \; \mathrm{P}(c_j) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j)$$

- Makes an overly strong independence assumption ---> easily yields class probabilities close to 1 or 0.
- However, classification accuracy is often surprisingly high (depends only on which class has the highest probability, not on how large the probability is)

- How to use unlabeled data: expectation-maximization algorithm
- Initialize parameters to Naive Bayes estimates from labeled documents only, then repeat E and M step convergence:
- E-step: calculate class label probabilities $\mathrm{P}(c_j|d_i)$ as above
- M-step: estimate

$$\mathrm{P}(w_t|c_j) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(c_j|d_i)} \qquad \mathrm{P}(c_j) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}|}$$

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: analyzed methods

- Co-training algorithm to be analyzed: (slightly different than the previous one)

- **Inputs:** An initial collection of labeled documents and one of unlabeled documents.
- Loop while there exist documents without class labels:
  - Build classifier A using the A portion of each document.
  - Build classifier B using the B portion of each document.
  - For each class C, pick the unlabeled document about which classifier A is most confident that its class label is C and add it to the collection of labeled documents.
  - For each class C, pick the unlabeled document about which classifier B is most confident that its class label is C and add it to the collection of labeled documents.
- **Output:** Two classifiers, A and B, that predict class labels for new documents. These predictions can be combined by multiplying together and then renormalizing their class probability scores.

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 1

- WebKB course webpage classification task: has natural feature split between page-content (bag of words) and links-to-page (bag of words from link anchor texts), but co-training does not actually help use unlabeled data more effectively than simple EM:

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	788	–0–	3.3%
Co-training	12	776	5.4%
EM	12	776	4.3%
Naive Bayes	12	–0–	13.0%

- Why does co-training not help?
  - Maybe the WebKB-Course task is too easy
  - Maybe the feature split is not independent enough (hyperlink text and page content talk about the same page)
  - Maybe co-training is unable to make good use of the independence between the feature split

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 2

- Artificially created data where feature sets are truly independent
- Four newsgroups from the "20 newsgroups" data set
- Positive examples: connect randomly picked documents from the first 2 newsgroups (1^{st} group = 1^{st} view, 2^{nd} group = 2^{nd} view)
- Negative examples: connect randomly picked documents from the last 2 newsgroups (3^{rd} group = 1^{st} view, 4^{th} group = 2^{nd} view)
- Use the same vocabulary in the 1^{st} and 3^{rd} groups, so that the 1^{st} view has the same feature space for both positive and negative examples; do the same for the 2^{nd} view
- Because the joined documents are randomly picked, both views are independent
- Each newsgroup is different ---> each view on its own should be enough to separate positive from negative examples

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

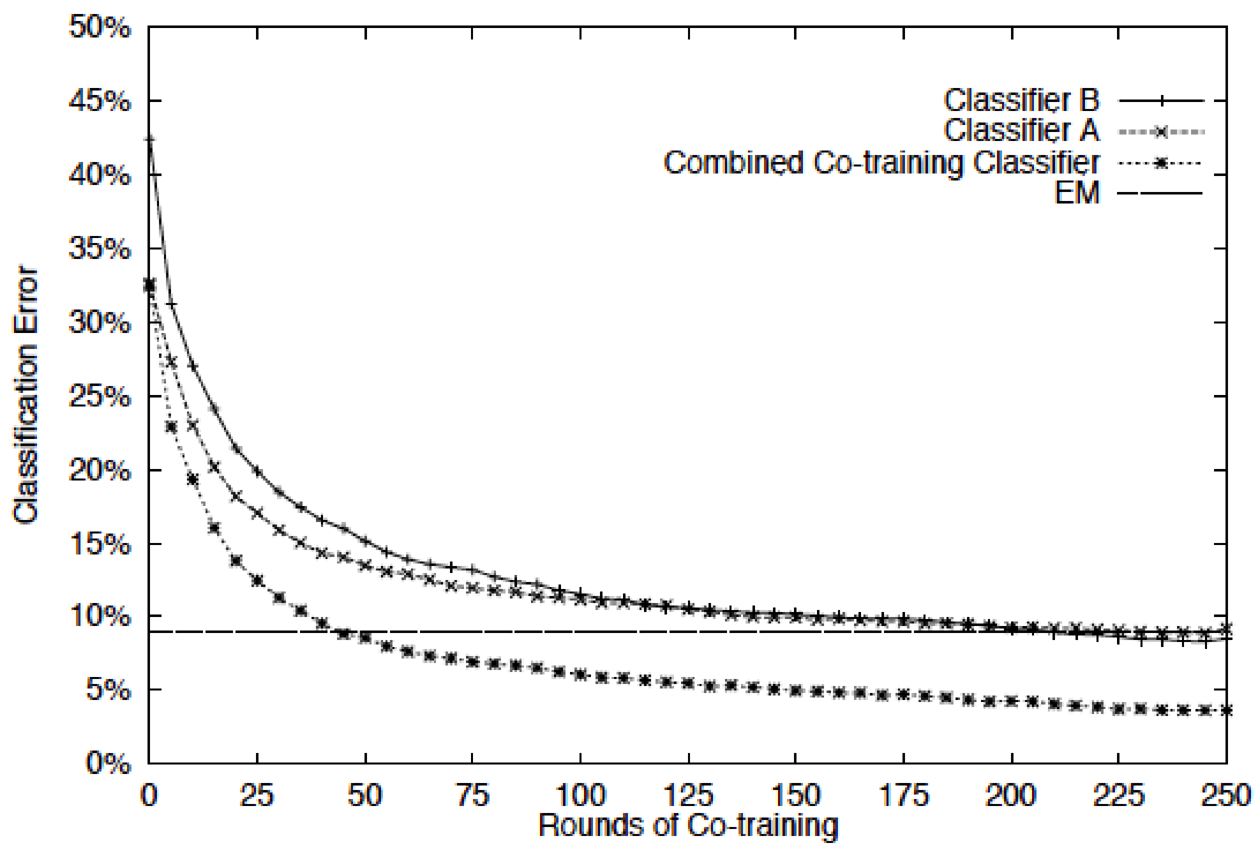
Co-training: test 2

Results: now co-training helps

Final performances

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	1006	–0–	3.9%
Co-training	6	1000	3.7%
EM	6	1000	8.9%
Naive Bayes	6	–0–	34.0%

Performance with iterations



Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 3, hybrid algorithms

- Create hybrid algorithms that are inbetween co-training and EM

- co-EM: iterative algorithm that uses the feature split.
    - initialize the A-feature-set naive Bayes classifier from labeled data
    - Label all unlabeled data probabilistically with A
    - Train classifier of B-featureset using labeled data and the unlabeled data with A's labels.
    - Relabel the data using B
    - Repeat the learning of A, then repeat learning of B, ..., until the classifiers converge
    - Combine the classifiers A and B by multiplying the class probabilities they give

•The co-EM uses one learner to give labels to all unlabeled data, and the second classifier learn from them; whereas co-training learns from only one example at a time

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 3, hybrid algorithms

- Create hybrid algorithms that are inbetween co-training and EM

- self-training: incremental algorithm that does not use the feature split (co-training setting).
    - Build a single naive Bayes classifier using labeled training data and all features
    - Label unlabeled data: convert the most confidently predicted document of each class into a labeled training example
    - Iterate until all unlabeled documents are given labels

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 3, hybrid algorithms

- Result:

Method	Uses Feature Split? Yes	No
Incremental	co-training	self-training
Iterative	co-EM	EM

Method	Uses Feature Split? Yes	No
Incremental	3.7%	5.8%
Iterative	3.3%	8.9%

- Algorithms that make use of an independent and redundant division of the features perform better than algorithms that do not

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Co-training: test 3, hybrid algorithms

• Why does co-training help? Hypothesis: because it is more robust to assumptions of its underlying classifiers.

• EM uses naive Bayes to assign posterior class probabilities to each unlabeled document. These probabilities can be poorly estimated because text data does not really have independent words.

• Co-training uses the classifieer to rank the documents by confidence, but does not use the actual posterior probabilities. This is a weaker use of the independence assumption than in EM (but still stronger then not using the assumption at all)

Following the approach from Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000. Images from that paper.

Part 3:

Bayesian Co-training

Bayesian co-training

• The previous discussion on co-training did not present it as an integrated model of data: does not optimize a joint cost function for all views of all data

• There have been approaches called **co-regularization** that do optimize a joint cost function (not discussed here) but they still optimize one view at a time

• Bayesian co-training: an undirected graphical model for co-training. Maximum likelihood inference for that model is related to co-regularization.

Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

Bayesian co-training

- We have m different views, *n* samples, sample *i* denoted as

$$\mathrm{x}_i \triangleq (\mathrm{x}_i^{(1)}, \ldots, \mathrm{x}_i^{(m)})$$

- All samples of a particular view j denoted as

$$\mathrm{x}^{(j)} \triangleq (\mathrm{x}_1^{(j)}, \ldots, \mathrm{x}_n^{(j)})$$

- Outputs of all samples denoted as $\mathrm{y} = [y_1, \ldots, y_n]^\top$

- We will use a Gaussian process based prediction for each view. Each view has an underlying function f that predicts the sample:

$$f_j \sim \mathcal{GP}(0, \kappa_j)$$

- The label y should depend on values of all the latent functions

Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

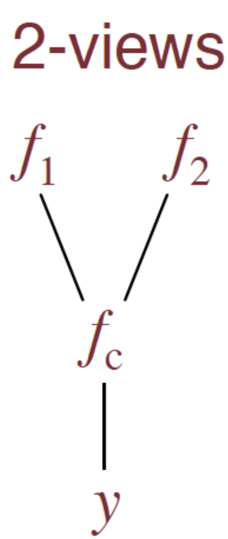
Bayesian co-training

- Idea: define a consensus function f_c that combines information from the individual functions, make the label depend on that alone.
- Joint distribution:

$$p(y, f_c, f_1, \ldots, f_m) = \frac{1}{Z}\psi(y, f_c)\prod_{j=1}^{m}\psi(f_j, f_c)$$

where ψ are some potential functions

- corresponding graphical model:



Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

Bayesian co-training

- For n samples:

$$p(\mathbf{y}, \mathbf{f}_c, \mathbf{f}_1, \dots, \mathbf{f}_m) = \frac{1}{Z} \prod_{i=1}^{n} \psi(y_i, f_c(\mathbf{x}_i)) \prod_{j=1}^{m} \psi(\mathbf{f}_j) \psi(\mathbf{f}_j, \mathbf{f}_c)$$

- The **within-view potential** and **consensus potential** can be defined as

$$\psi(\mathbf{f}_j) = \exp\left(-\frac{1}{2}\mathbf{f}_j^\top \mathbf{K}_j^{-1} \mathbf{f}_j\right), \quad \psi(\mathbf{f}_j, \mathbf{f}_c) = \exp\left(-\frac{\|\mathbf{f}_j - \mathbf{f}_c\|^2}{2\sigma_j^2}\right)$$

$$\mathbf{K}_j(\mathbf{x}_k, \mathbf{x}_\ell) = \kappa_j(\mathbf{x}_k^{(j)}, \mathbf{x}_\ell^{(j)})$$

- The **output potential** is

$$\psi(y_i, f(\mathbf{x}_i)) = \begin{cases} \exp(-\frac{1}{2\sigma^2}\|y_i - f(\mathbf{x}_i)\|^2) & \text{for regression,} \\ \lambda(y_i f(\mathbf{x}_i)) & \text{for classification.} \end{cases}$$

Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

Bayesian co-training

- When some samples are unlabeled, the likelihood becomes:

$$p\left(\mathrm{y}_l, \mathrm{f}_c, \mathrm{f}_1, \ldots, \mathrm{f}_m\right) = \frac{1}{Z} \prod_{i=1}^{n_l} \psi(y_i, f_c(\mathbf{x}_i)) \prod_{j=1}^{m} \psi(\mathrm{f}_j) \psi(\mathrm{f}_j, \mathrm{f}_c).$$

- Inference: try to integrate out the functions, at least the consensus function. It can be shown

$$p(\mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \exp\left\{ -\frac{1}{2} \sum_{j=1}^{m} \mathrm{f}_j^\top \mathrm{K}_j^{-1} \mathrm{f}_j - \frac{1}{2} \sum_{j<k} \left[\frac{\|\mathrm{f}_j - \mathrm{f}_k\|^2}{\sigma_j^2 \sigma_k^2} \bigg/ \sum_{\ell} \frac{1}{\sigma_\ell^2} \right] \right\}$$

- and for regression

$$p(\mathrm{y}, \mathrm{f}_1, \ldots, \mathrm{f}_m) = \frac{1}{Z} \exp\left\{ -\frac{1}{2\rho\sigma^2} \sum_j \frac{\sum_{i=1}^{n} (y_i - f_j(\mathbf{x}_i))^2}{\sigma_j^2} \right.$$
$$\left. - \frac{1}{2} \sum_j \mathrm{f}_j^\top \mathrm{K}_j^{-1} \mathrm{f}_j - \frac{1}{2\rho} \sum_{j<k} \frac{\|\mathrm{f}_j - \mathrm{f}_k\|^2}{\sigma_j^2 \sigma_k^2} \right\}$$

- Inference proceeds from there, more information on next lecture

Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

Bayesian co-training

- Example, result, details on next lecture:
  citeseer data (scientific papers in 6 classes)

	# Train +2/-10		# Train +4/-20	
Model	AUC	F1	AUC	F1
Text	0.5725 ± 0.0180	0.1359 ± 0.0565	0.5770 ± 0.0209	0.1443 ± 0.0705
Inbound Link	0.5451 ± 0.0025	0.3510 ± 0.0011	0.5479 ± 0.0035	0.3521 ± 0.0017
Outbound Link	0.5550 ± 0.0119	0.3552 ± 0.0053	0.5662 ± 0.0124	0.3600 ± 0.0059
Text+Link	0.5730 ± 0.0177	0.1386 ± 0.0561	0.5782 ± 0.0218	0.1474 ± 0.0721
Co-Trained GPLR	0.6459 ± 0.1034	0.4001 ± 0.2186	0.6519 ± 0.1091	0.4042 ± 0.2321
Bayesian Co-Training	$\mathbf{0.6536 \pm 0.0419}$	$\mathbf{0.4210 \pm 0.0401}$	$\mathbf{0.6880 \pm 0.0300}$	$\mathbf{0.4530 \pm 0.0293}$

Following the approach from Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. Bayesian Co-Training. Journal of Machine Learning Research 12:2649-2680, 2011. Images from that paper.

References

- Blum, Mitchell. **Combining labeled and unlabeled data with co-training.** In proceedings of COLT 1998.
- Nigam, K. and Ghani, R. **Analyzing the effectiveness and applicability of co-training.** In Proceedings of 9th international conference on information and knowledge management, pp. 86-93, 2000.
- Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, R. Bharat Rao. **Bayesian Co-Training.** Journal of Machine Learning Research 12:2649-2680, 2011.
- Minmin Chen, Kilian Q. Weinberger, John Blitzer. **Co-Training for Domain Adaptation.** Proceedings of NIPS 2011.