Active + Semi-Supervised Learning = Robust Multi-View Learning

Ion Muslea MUSLEA@ISI.EDU
Information Sciences Institute / University of Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292, USA

Steven Minton MINTON@FETCH.COM
Fetch Technologies, 4676 Admiralty Way, Marina del Rey, CA 90292, USA

Craig A. Knoblock KNOBLOCK@ISI.EDU
Information Sciences Institute / University of Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292, USA

Abstract

In a *multi-view* problem, the features of the domain can be partitioned into disjoint subsets (*views*) that are sufficient to learn the target concept. Semi-supervised, multi-view algorithms, which reduce the amount of labeled data required for learning, rely on the assumptions that the views are *compatible* and *uncorrelated* (i.e., every example is identically labeled by the target concepts in each view; *and*, given the label of any example, its descriptions in each view are independent). As these assumptions are unlikely to hold in practice, it is crucial to understand the behavior of multi-view algorithms on problems with incompatible, correlated views. We address this issue by studying several algorithms on a parameterized family of text classification problems in which we control both view correlation and incompatibility. We first show that existing semi-supervised algorithms are not robust over the whole spectrum of parameterized problems. Then we introduce a new multi-view algorithm, Co-EMT, which combines semi-supervised and active learning. Co-EMT outperforms the other algorithms both on the parameterized problems and on two additional real world domains. Our experiments suggest that Co-EMT's robustness comes from active learning compensating for the correlation of the views.

1. Introduction

In a multi-view problem, one can partition the domain's features in subsets that are *sufficient* for learning the target concept. For instance, as described by Blum and Mitchell (1998), one can classify segments of televised broadcast based *either* on the video *or* on the audio information; or one can classify Web pages based on the words that appear *either* in the documents *or* in the hyperlinks pointing to them. In this paper we focus on two types of multi-view algorithms that reduce the amount of labeled data required for learning: *semi-supervised* and *active learning* algorithms. The former type bootstraps the views from each other in order to boost the accuracy of a classifier learned based on a few labeled examples. The latter detects the most informative unlabeled examples and asks the user to label them. Both types of multi-view algorithms have been applied to a variety of real-world domains, from natural language processing (Collins & Singer, 1999) and speech recognition (de Sa & Ballard, 1998) to information extraction (Muslea et al., 2000).

The theoretical foundations of multi-view learning (Blum & Mitchell, 1998) are based on the assumptions that the views are both *compatible* and *uncorrelated*. Intuitively, a problem has *compatible* views if all examples are labeled identically by the target concepts in each view. On the other hand, two views are *uncorrelated* if, given the label of any example, its descriptions in each view are independent. In real-world problems, both assumptions are likely to be violated for a variety of reasons (e.g, correlated or insufficient features). Consequently, in this paper we study the robustness of multi-view algorithms with respect to view incompatibility and correlation. As in practice it is difficult to measure these two factors, we use in our study a parameterized family of text classification problems in which we control both view incompatibility and correlation.

In our empirical investigation we consider four algorithms: semi-supervised EM (Nigam et al., 2000), Co-Training (Blum & Mitchell, 1998), Co-EM (Nigam & Ghani, 2000), and Co-EMT. The first three are semi-supervised algorithms that were successfully applied to text classification problems. Finally, Co-EMT is a new multi-view algorithm that interleaves active and semi-supervised learning; that is, Co-EMT uses a multi-view active learning algorithm, Co-

Testing (Muslea et al., 2000), to select the labeled examples for the multi-view, semi-supervised Co-EM.

Our experiments lead to two important conclusions. First, Co-EMT clearly outperforms the other three algorithms in the entire `correlation - incompatibility` space. These results obtained on the parameterized problems are further reinforced by experiments on two additional real world domains. Second, the robustness of Co-EMT is due to active learning compensating for view correlation.

2. Issues in the Multi-View Setting

The *multi-view setting* (Blum & Mitchell, 1998) applies to learning problems that have a natural way to divide their features into subsets (*views*) each of which are *sufficient* to learn the target concept. In such problems, an example x is described by a different set of features in each view. For example, in a domain with two views **V1** and **V2**, any example x can be seen as a triple $[x_1, x_2, l]$, where x_1 and x_2 are its descriptions in the two views, and l is its label.

Blum and Mitchell (1998) proved that for a problem with two views the target concept can be learned based on a few labeled and many unlabeled examples, provided that the views are *compatible* and *uncorrelated*. The former condition requires that all examples are labeled identically by the target concepts in each view. The latter means that for any example $[x_1, x_2, l]$, x_1 and x_2 are independent given l.

The proof in (Blum & Mitchell, 1998) is based on the following argument: one can learn a weak hypothesis h_1 in **V1** based on the few labeled examples and then apply h_1 to all unlabeled examples. If the views are uncorrelated, these newly labeled examples are seen in **V2** as a random training set with classification noise, based on which one can learn the target concept in **V2**. Both the requirements that the views are compatible and uncorrelated are crucial in this process.[1] If the views are correlated, the training set in **V2** is *not* random; if the views are incompatible, the target concepts in the two views label a large number of examples *differently*. Consequently, from **V2**'s perspective, h_1 may "misslabel" so many examples that learning the target concept in **V2** becomes impossible.

To introduce the intuition behind view incompatibility and correlation, let us consider the COURSES problem (Blum & Mitchell, 1998), in which Web pages are classified as "*course homepages*" and "*other pages*." The views **V1** and **V2** consist of words in the hyperlinks pointing to the pages and words in the Web pages, respectively. Figure 1

[1] An updated version of (Blum & Mitchell, 1998) shows that the theoretical guarantees also hold for partially incompatible views, provided that they are uncorrelated. However, in practice one cannot ignore view incompatibility because one rarely, if ever, encounters real world problems with uncorrelated views.

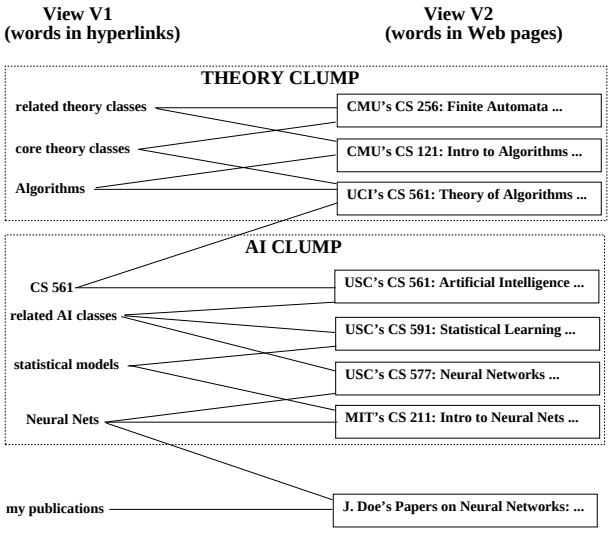

Figure 1. **Two illustrative clumps in the** COURSES **domain.**

shows several illustrative examples for the COURSES problem. Each of the **17 lines** in Figure 1 represents an example; that is, we depict each example x as a **line** that connects its descriptions x_1 and x_2 in the two views. All but the two bottom examples (i.e., **lines**) are "*course homepages*"; consequently, to keep Figure 1 simple, we do not show the examples' labels. Note that in Figure 1 the *same page* may be referred by several hyperlinks, while several hyperlinks that contain the *same text* may point to different pages.

In real world problems, the views are partially incompatible for a variety of reasons: corrupted features, insufficient attributes, etc. For instance, as shown in Figure 1, of the three hyperlinks that contain the text "Neural Nets", two point to homepages of neural nets classes, while the third one points to a publications page. That is, Web pages with different labels in **V2** have the *same* description in **V1**. Consequently, ["Neural Nets", "MIT's CS 211: ..."] and ["Neural Nets", "J. Doe's Papers ...] are incompatible because they require that "Neural Nets" has simultaneously two different labels.

In practice, the views are also (partially) correlated because of *domain clumpiness*, which can be best introduced by an example. Consider, for instance, the eight multi-view examples of AI homepages that are *depicted as lines* within the "AI CLUMP" rectangle in Figure 1. We call such a group of examples a *clump* because the bi-partite subgraph that has as vertices the four hyperlinks and four Web pages, respectively, is heavily connected by the eight edges representing the examples. Note that two clumps per class are sufficient to violate the "uncorrelated views" assumption: for any example x, it is highly likely that its descriptions in the two views come from the same clump. Intuitively, this means that it is unlikely to encounter examples such as ["CS 561", "UCI's CS 561: Theory of Algorithms"], which connects the THEORY and AI clumps (see Figure 1).

Given:
- a learning problem with two views **V1** and **V2**
- a learning algorithm $\mathcal{L}$
- the sets T and U of labeled and unlabeled examples
- the number k of iterations to be performed

Co-Training:
LOOP for k iterations
- use $\mathcal{L}$, **V1**(T), and **V2**(T) to create classifiers h_1 and h_2
- FOR EACH class C_i DO
  - let $E1$ and $E2$ be the e unlabeled examples on which h_1 and h_2 make the most confident predictions for C_i
  - remove $E1$ and $E2$ from U, label them according to h_1 and h_2,respectively, and add them to T
- combine the prediction of h_1 and h_2

Semi-supervised EM:
- let $All = T \bigcup U$
- let h be the classifier obtained by training $\mathcal{L}$ on T
LOOP for k iterations
- New = ProbabilisticallyLabel(All, h)
- $h = \mathcal{L}_{MAP}(New)$

Co-EM:
- let $All = T \bigcup U$
- let h_1 be the classifier obtained by training $\mathcal{L}$ on T
LOOP for k iterations
- New_1 = ProbabilisticallyLabel(All, h_1)
- $h_2 = \mathcal{L}_{MAP}$(**V2**(New_1))
- New_2 = ProbabilisticallyLabel(All, h_2)
- $h_1 = \mathcal{L}_{MAP}$(**V1**(New_2))
- combine the prediction of h_1 and h_2

Figure 2. **Co-Training, Semi-supervised EM, and Co-EM.**

3. Semi-supervised Algorithms

In this section we provide a high-level description of the semi-supervised algorithms that are used in our comparison: Co-Training, semi-supervised EM, and Co-EM.

3.1 Co-Training

Co-Training (Blum & Mitchell, 1998) is a semi-supervised, multi-view algorithm that uses the initial training set to learn a (weak) classifier in each view. Then each classifier is applied to all unlabeled examples, and Co-Training detects the examples on which each classifier makes the most confident predictions. These high-confidence examples are labeled with the estimated class labels and added to the training set (see Figure 2). Based on the new training set, a new classifier is learned in each view, and the whole process is repeated for several iterations. At the end, a final hypothesis is created by a voting scheme that combines the prediction of the classifiers learned in each view.

3.2 Semi-supervised EM

Semi-supervised EM (Nigam & Ghani, 2000) is a single-view algorithm that we use as baseline. As shown in Figure 2, it applies a probabilistic learning algorithm $\mathcal{L}$ to a small set of labeled examples and a large set of unlabeled ones. First, semi-supervised EM creates an initial classifier h based solely on the labeled examples. Then it repeatedly performs a two-step procedure: first, use h to probabilistically label all unlabeled examples; then, learn a new *maximum a posteriori* (MAP) hypothesis h based on the examples labeled in the previous step. Intuitively, EM tries to find the most likely hypothesis that could generate the distribution of the unlabeled data. Semi-supervised EM can be seen as clustering the unlabeled data "around" the examples in the original training set.

3.3 Co-EM

Co-EM(Nigam & Ghani, 2000) is a semi-supervised, multi-view algorithm that uses the hypothesis learned in one view to probabilistically label the examples in the other one (see Figure 2). Intuitively, Co-EM runs EM in each view and, before each new EM iteration, inter-changes the probabilistic labels generated in each view.

Co-EM can be seen as a probabilistic version of Co-Training. In fact, both algorithms are based on the same underlying idea: they use the knowledge acquired in one view (i.e., the probable labels of the examples) to train the other view. The major difference between the two algorithms is that Co-EM does *not* commit to a label for the unlabeled examples; instead, it uses probabilistic labels that may change from one iteration to the other. [2] By contrast, Co-Training's commitment to the high-confidence predictions may add to the training set a large number of mislabeled examples, especially during the first iterations, when the hypotheses may have little prediction power.

3.4 An Empirical Comparison

In this section, we motivate the need for a new, robust multi-view algorithm by showing that existing algorithms have an uneven performance in different regions of the correlation - incompatibility space. For this purpose, we compare EM, Co-Training, and Co-EM on a parameterized family of problems for which we control the level of clumpiness (one, two, and four clumps per class) and incompatibility (0%, 10%, 20%, 30%, and 40% of the examples are incompatible). To keep the presentation succinct, we present here only the information critical to making our case. The experimental framework and the complete results are presented in detail in Section 5; the parameterized family of problems is discussed in Appendix A.

[2] In (Nigam & Ghani, 2000), Co-EM and Co-Training are contrasted as being *iterative* and *incremental*, respectively. This description is equivalent to ours: Co-EM *iteratively* uses the unlabeled data because it *does not commit* to the labels from the previous iteration. By contrast, Co-Training *incrementally* uses the unlabeled data by *committing* to a few labels per iteration.

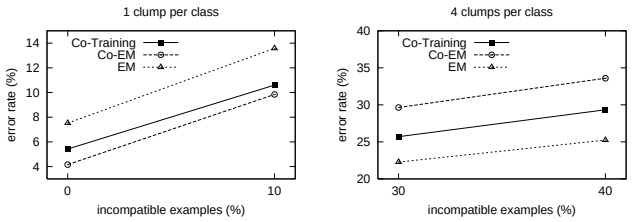

Figure 3. **A comparison of the semi-supervised algorithms.**

In Figure 3 we show the performance of EM, Co-Training, and Co-EM in two regions of the `correlation - incompatibility` space. In the graph on the left, the algorithms are compared on problems with uncorrelated views (one clump per class) that are highly compatible (0% and 10% of the examples are incompatible). In the second graph, the algorithms are applied to problems with highly incompatible views (30% and 40% of the examples are incompatible) that have four clumps per class. The `x` axis shows the percentage of incompatible examples in the problems, while the `y` axis represents the error rates.

These results show that the three algorithms are sensitive to view incompatibility and correlation. For example, Co-EM and Co-Training outperform EM on problems with highly compatible, uncorrelated views. In contrast, as the views become correlated and incompatible, the two multi-view algorithms underperform EM, with Co-EM doing clearly worse than Co-Training. In the next section, we introduce a new algorithm, Co-EMT, that has a robust behavior over the entire spectrum of problems.

4. Co-Testing + Co-EM = Co-EMT

Co-Testing (Muslea et al., 2000) is a family of multi-view active learning algorithms that start with a few labeled examples and a pool of unlabeled ones. Co-Testing searches for the most informative examples in the unlabeled pool and asks the user to label them. As shown in Figure 4, Co-Testing repeatedly trains one hypothesis for each view and queries one of the unlabeled examples on which the two hypotheses predict different labels (also called *contention points*). Intuitively, if two *compatible* views disagree about a label, at least one of them must be wrong. Consequently, by asking the user to label a contention point, Co-Testing provides useful information for the view that mislabeled it.

Co-EMT is a novel algorithm that interleaves Co-EM and Co-Testing (see Figure 4).[3] As opposed to a typical Co-Testing algorithm, which learns h_1 and h_2 based solely on labeled examples, Co-EMT induces the two hypotheses by running Co-EM on both labeled and unlabeled examples.

[3] In this paper we have chosen to combine Co-Testing with Co-EM rather than Co-Training because of the difficulties encountered while fine-tuning the latter, which is sensitive to changes in the number of examples added after each iteration.

`Given:`
- a learning problem with two views **V1** and **V2**
- a learning algorithm $\mathcal{L}$
- the sets T and U of labeled and unlabeled examples
- the number N of queries to be made

Co-Testing:
`REPEAT` N times
- use $\mathcal{L}$, **V1**(T), and **V2**(T) to create classifiers h_1 and h_2
- let $ContentionPoints = \{x \in U, h_1(x) \neq h_2(x)\}$
- select query among $ContentionPoints$ & ask user to label it
- move newly-labeled contention point from U to T

- combine the prediction of h_1 and h_2

Co-EMT:
- let $iters$ be the number of Co-EM iterations within Co-EMT

`REPEAT` N times
- run **Co-EM**($\mathcal{L}$, **V1**, **V2**, T, U, $iters$) to learn h_1 and h_2
- let $ContentionPoints = \{x \in U, h_1(x) \neq h_2(x)\}$
- select query among $ContentionPoints$ & ask user to label it
- move newly-labeled contention point from U to T

- combine the prediction of h_1 and h_2

Figure 4. **The Co-Testing and Co-EMT Algorithms.**

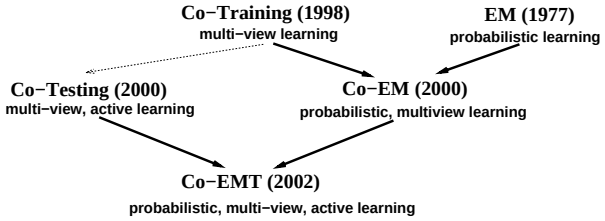

Figure 5. **The lineage of the Co-EMT algorithm.**

In the current implementation, Co-EMT uses a straightforward query selection strategy: it asks the user to label the contention point on which the combined prediction of h_1 and h_2 is the least confident (i.e., it queries one of the unlabeled examples on which h_1 and h_2 have an *equally strong confidence* at predicting a *different label*).

In order to put Co-EMT in a larger context, in Figure 5 we show its relationship with the other algorithms considered in this study. One one side, Co-EMT is a semi-supervised variant of Co-Testing, which - in turn - was inspired from Co-Training. On the other side, Co-EMT builds on Co-EM, which is a state-of-the art, semi-supervised algorithm that combines the basic ideas from Co-Training and EM.

Note that interleaving Co-EM and Co-Testing leads to an interesting synergy. On one hand, Co-Testing boosts the accuracy of Co-EM by selecting a highly informative set of labeled examples (stand-alone Co-EM chooses them at random). On the other hand, as the hypotheses learned by Co-EM are more accurate than the ones learned just from labeled data, compared with stand-alone Co-Testing, Co-EMT uses more accurate hypotheses to select the queries.

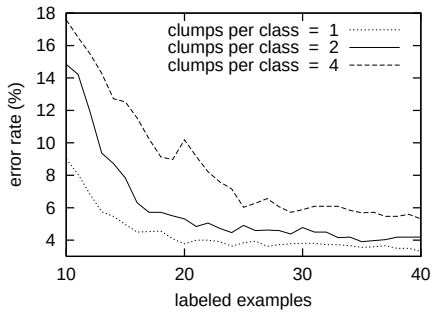

Figure 6. **Illustrative learning curves for Co-EMT on tasks with no incompatibility and 1, 2, and 4 clumps per class.**

5. Empirical Results

5.1 The Experimental Setup

In our empirical investigation, we apply EM, Co-Training, Co-EM, Co-Testing, and Co-EMT on a family of problems in which we control both the clumpiness and the view incompatibility. We have created problems with one, two, and four clumps per class. For each level of clumpiness, we have generated problems with 0%, 10%, 20%, 30%, and 40% incompatible examples. For each of these 15 points in the `correlation - incompatibility` space, we have created four text classification problems, for a total of 60 problems (see Appendix **A** for details).

The accuracy of the algorithms is estimated based on four runs of 5-fold cross-validation; consequently, each training and test set consist of 640 and 160 examples, respectively. For the three semi-supervised algorithms, the 640 training examples are split randomly into two groups: 40 of them are used as labeled examples, while the remaining 600 are unlabeled (i.e., we hide their labels). To keep the comparison fair, Co-EMT and Co-Testing start with 10 randomly chosen labeled examples and query 30 of the 630 unlabeled ones, for a total of 40 labeled examples (see Figure 6 for three illustrative learning curves).

We use Naive Bayes as the underlying algorithm $\mathcal{L}$. For EM, Co-Training, Co-EM, and Naive Bayes, we have implemented the versions described in (Nigam & Ghani, 2000). EM and Co-EM are run for seven and five iterations, respectively. Co-Training, which require significant fine tuning, labels 40 examples after each of the seven iterations. To avoid prohibitive running time, within Co-EMT, we perform only two Co-EM iterations after each Co-Testing query (on each of the 60 problems, Co-EMT runs Co-EM after each of the 600 queries: 4 runs $\times$ 5 folds $\times$ 30 queries per fold). At each point in the `correlation - incompatibility` space, the reported error rate is averaged over four text classification problems.

Figure 7 shows the performance of Co-EMT, Co-Testing, Co-EM, Co-Training, and EM on the parameterized family of problems. The five graphs correspond to the five levels of views incompatibility: 0%, 10%, 20%, 30%, and 40%. In each graph, the `x` and `y` axes show the number of clumps per class and the error rate, respectively.

Co-EMT obtains the lowest error rates on **all** 15 points in the `correlation-incompatibility` space. In a pairwise comparison with Co-Testing, Co-Training, Co-EM, and EM, our results are statistically significant with 95% confidence on 15, 13, 10, and 12 of the points. The remaining points represent "extreme situations" that are unlikely to occur in practice: for Co-Training and Co-EM, conditional independent views (one clump per class); for EM highly correlated and incompatible views (four clumps per class, and 20%, 30%, 40% incompatibility).

5.2 Discussion

These empirical results deserve several comments. First, Co-EMT, which combines Co-Testing and Co-EM, clearly outperforms both its components. Intuitively, Co-EMT's power comes from Co-Testing and Co-EM compensating for each other's weaknesses. On one hand, by exploiting the unlabeled data, Co-EM boosts the accuracy of the classifiers learned by Co-Testing. On the other hand, Co-Testing improves Co-EM's accuracy by providing a highly informative set of labeled examples.

Co-EMT is not the first algorithm that combines semi-supervised and active learning: in (McCallum & Nigam, 1998b), various combinations of semi-supervised EM and Query-by-Committee (QBC) are shown to outperform both EM and QBC.[4] We expect that using other active learning algorithms to select the labeled examples for Co-EM, Co-Training, and EM would also improve their accuracy. Finding the best combination of active and semi-supervised learning is beyond the scope of this paper. Our main contribution is to show that interleaving active and semi-supervised learning leads to a robust performance over the entire spectrum of problems.

Second, Co-EM and Co-Training are highly sensitive to domain clumpiness. On problems with uncorrelated views (i.e., one clump per class), Co-EM and Co-Training clearly outperform EM. In fact, Co-EM is so accurate that Co-EMT can barely outperform it. This behavior is consistent with theoretical argument in (Blum & Mitchell, 1998):

[4] The best of these EM and QBC combinations is not appropriate for multi-view problems because it uses a sophisticated heuristic that estimates the density of various regions in the *single-view* instance space (the density of a multi-view instance space is a function of the "local" densities within each view). Instead, we have implemented another (single-view) algorithm from (McCallum & Nigam, 1998b), which, similarly to Co-Testing, interleaves QBC and EM. As this algorithm barely improved EM's accuracy on the parameterized problems, we decided not to show the corresponding learning curves on the already crowded Figure 7.

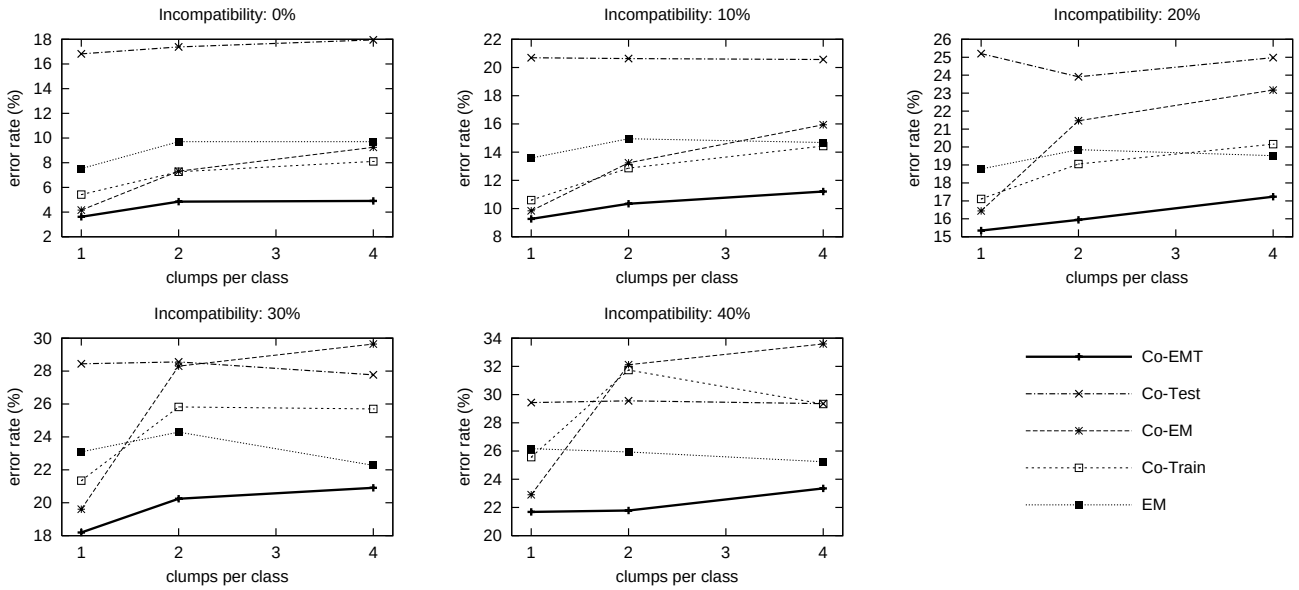

Figure 7. **Results on the parameterized family of problems.**

given two uncorrelated views, even in the presence of view incompatibility, a concept can be learned based on a few labeled and many unlabeled examples.

In contrast, on problems with four clumps per class, EM clearly outperforms both Co-EM and Co-Training. The two multi-view algorithms perform poorly on clumpy domains because rather than being disseminated over the entire instance space, the information exchanged between the views remains localized within each clump. The fact that Co-EMT is almost insensitive to clumpiness suggests that Co-Testing compensates for domain clumpiness.[5]

Third, the performance of all algorithms degrades as the views become less compatible. The multi-view algorithms are sensitive to view incompatibility because the information exchanged between views becomes misleading as more examples are labeled differently in the two views. To cope with this problem, in a companion paper (Muslea et al., 2002) we introduce a *view validation* technique that detects whether or not two views are "sufficiently compatible" for multi-view learning.

Note that, at first glance, Co-EMT should perform poorly on problems with highly incompatible views: on such domains, it looks likely that Co-EMT will query incompatible examples, which convey little information and are misleading for Co-EM. To understand how Co-EMT avoids making such queries, let us reconsider the situation in Section 2, where two hyperlinks containing the either *same text* ("Neural Nets") or *similar* fragments of text (e.g., "Artificial Neural Nets" and "Artificial Neural Networks") can point to Web pages having *different labels*. Because of the ambiguity of such examples, the hypotheses learned in the "hyperlink view" have a *low confidence* in predicting their labels. As Co-EMT queries contention points on which the views make equally confident predictions, it follows that an incompatible example is queried only if the other view also has an equally low confidence on its prediction.

In summary, we expect Co-EMT to perform well on most domains. The areas of the `correlation - incompatibility` space in which it does not clearly outperform all other four algorithms have either uncorrelated views (one clump per class) or correlated, incompatible views (four clumps per class, 30%-40% incompatibility). On the former it barely outperforms Co-EM, but such problems are unlikely to occur in practice. On the latter it barely outperforms EM, and one may expect EM to outperform Co-EMT at higher incompatibility levels. To cope with this problem, we use *view validation* (Muslea et al., 2002) to predict whether two views are sufficiently compatible for learning.

5.3 Results on real-world problems

In order to strengthen the results obtained on the parameterized family of problems, we present now an additional experiment on two real-world domains: COURSES (Blum & Mitchell, 1998) and ADS (Kushmerick, 1999). In COURSES (1041 examples), we classify Web pages as course homepages or not. The two views consist of words that appear in the pages and in the hyperlinks pointing to them, respectively. In ADS (3279 examples), we classify images that appear in Web pages as ads or non-ads. One view describes

[5] Remember that Co-EMT is simply Co-EM using labeled examples chosen via Co-Testing queries.

Algorithm	COURSES	ADS
Co-EMT	**3.98± 0.6**	**5.75±0.4**
Co-Testing	4.80 ± 0.5	7.70 ± 0.4
Co-EM	5.08 ± 0.7	7.80 ± 0.4
EM	5.32 ± 0.6	8.55 ± 0.4
Co-Training	5.18 ± 0.6	7.54 ± 0.4

Table 1. **Error rates on two real world problems.**

the image itself (e.g., words in the image's URL and caption), while the other view characterizes related pages (e.g., words from the URLs to the pages that contain the image or are pointed-at by the image). [6] For both domains we perform two runs of 5-fold cross validation. On COURSES, the Co-EM, Co-Training, and EM use 65 labeled examples, while Co-EMT and Co-Testing start with 10 labeled examples and make 55 queries. For ADS, the semi-supervised algorithms use 100 labeled examples, while Co-EMT and Co-Testing start with 60 labeled examples and make 40 queries. EM, Co-EM and Co-Training are run for seven, five and four iterations, respectively (Co-Training adds 100 examples after each iteration). Finally, within Co-EMT, we perform two Co-EM iterations after each Co-Testing query.

Table 1 shows that Co-EMT again obtains the best accuracy of the five algorithms. Except for the comparison with Co-Tesing and Co-EM on COURSES, the results are statistically significant with at least 95% confidence.

6. Conclusions and Future Work

In this paper we used a family of parameterized problems to analyze the influence of view correlation and incompatibility on the performance of several multi-view algorithms. We have shown that existing algorithms are not robust over the whole `correlation - incompatibility` space. To cope with this problem, we introduced a new multi-view algorithm, Co-EMT, that interleaves active and semi-supervised learning. We have shown that Co-EMT clearly outperforms the other algorithms both on the parameterized problems and on two real world domains. Our experiments suggest that the robustness of Co-EMT comes from active learning compensating for the view correlation.

We plan to continue our work along two main directions. First, we intend to study other combinations of Co-Testing and semi-supervised algorithms, both on semi-artificial and real-world domains. In particular, we plan to use *multiple mixture components* (Nigam et al., 2000) to model and cope with domain clumpiness (i.e., to automatically generate a component for each clump in a class). Second, we intend to work on the *view detection* problem, in which one tries to detect the existence of multiple views within a given domain. We plan to generate several *candidate views* (i.e., features partitions) and to use *view validation* (Muslea et al., 2002) to predict whether the views are appropriate for multi-view learning.

[6] As all features in ADS are *boolean* (i.e., presence/absence of word in document), we use Naive Bayes with the multi-variate Bernoulli model(McCallum & Nigam, 1998a).

References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proc. of the Conference on Computational Learning Theory* (pp. 92–100).

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Proc. of the Empirical NLP and Very Large Corpora Conference* (pp. 100–110).

de Sa, V., & Ballard, D. (1998). Category learning from multi-modality. *Neural Computation, 10*, 1097–1117.

Joachims, T. (1996). A probabilistic analysis of the ROCCHIO algorithm with TFIDF for text categorization. *Computer Science Tech. Report CMU-CS-96-118*.

Kushmerick, N. (1999). Learning to remove internet advetisements. *Proc. of Auton. Agents-99* (pp. 175–181).

McCallum, A., & Nigam, K. (1998a). A comparison of event models for naive bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*.

McCallum, A., & Nigam, K. (1998b). Employing EM in pool-based active learning for text classification. *Proc. of Intl. Conference on Machine Learning* (pp. 359–367).

Muslea, I., Minton, S., & Knoblock, C. (2000). Selective sampling with redundant views. *Proc. of National Conference on Artificial Intelligence* (pp. 621–626).

Muslea, I., Minton, S., & Knoblock, C. (2002). Adaptive view validation: A case study on wrapper induction. *To appear in Proc. of ICML-2002*.

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proc. of Information and Knowledge Management* (pp. 86–93).

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

A. The 60 Semi-Artificial Problems

To create a parameterized set of problems in which we control the view correlation and incompatibility, we generalize an idea from (Nigam & Ghani, 2000). One can create a (semi-artificial) domain with compatible, uncorrelated views by taking two *unrelated* binary classification problems and considering each problem as an individual view.

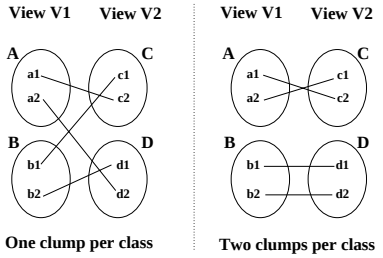

Figure 8. **Generating one and two clumps per class.**

The multi-view examples are created by randomly pairing examples that have the same label in the original problems.

The procedure above can be easily modified to introduce both clumps and incompatible examples. For instance, consider creating a binary classification problem in which the positive examples consist of two clumps. We begin with *four* unrelated problems that have the sets of positive examples *A*, *B*, *C*, and *D*, respectively. In the newly created 2-view problem, the positive examples in the views **V1** and **V2** consist of the $A \bigcup B$ and $C \bigcup D$, respectively. As shown in the left-most graph in Figure 8, if the multi-view examples are created by randomly pairing an example from $A \bigcup B$ with one from $C \bigcup D$, we obtain, again, uncorrelated views. By contrast, if we allow the examples from A to be paired only with the ones from C, and the ones from B with the ones from D, we obtain a problem with two clumps of positive examples: A-C and B-D. Similarly, based on eight or 16 unrelated problems, one can create four or eight clumps per class, respectively.

Adding incompatible examples is a straightforward task: first, we randomly pick one positive and one negative multi-view example, say ["Intro to AI", AI-Class] and ["J. Doe", JDoe-Homepage]. Then we replace these two examples by their "recombinations": the "positive" example ["Intro to AI", JDoe-Homepage] and the "negative" example ["J. Doe", AI-Class]. Note that the labels of the two new examples are correct in one view (the hyperlink words) and incorrect in the other one (the words in the page). In this context, a level of, say, 40% incompatibility means that 40% of the examples in both the training and the test set are assigned a label that is correct only in one of the views. Similarly, when Co-EMT queries an incompatible example, we provide the label that is correct only in one of the views.

In order to generate problems with up to four clumps per class, we used 16 of 20 newsgroups postings from the Mini-Newsgroups dataset,[7] which is a subset of the well-known `20-Newsgroups` domain (Joachims, 1996). Each newsgroup consists of 100 articles that were randomly chosen from the 1000 postings included in the original dataset. We divided the 16 newsgroups in four groups of four (see Table 2). The examples in each such group are used as either

[7] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/mini_newsgroups.tar.gz

	V1	**V2**
pos	comp.os.ms-win.misc comp.sys.ibm.pc.hrwd rec.autos rec.sport.baseball	comp.windows.x comp.sys.mac.hrwd rec.motorcycles rec.sport.hockey
neg	sci.crypt sci.space talk.politics.guns talk.politics.misc	sci.electronics sci.med talk.politics.mideast talk.religion.misc

Table 2. **The 16 newsgroups included in the domain.**

positive or negative examples in one of the two views; i.e., the newsgroups `comp.os.ms-win`, `comp.sys.ibm`, `comp.windows.x`, and `comp.sys.mac` play the roles of the *A*, *B*, *C*, and *D* sets of examples from Figure 8.

We begin by creating *compatible* views with three levels of clumpiness: one, two, and four clumps per class. For one clump per class, any positive example from **V1** can be paired with any positive example in **V2**. For two clumps per class, we do *not* allow the pairing of `comp` examples in one view and the `rec` examples in the other one. Finally, for four clumps per class we pair examples from `comp.os.ms-win` and `comp.windows.x`, from `comp.sys.ibm` and `comp.sys.mac`, etc.

For each level of clumpiness, we consider with five levels of view incompatibility: 0%, 10%, 20%, 30%, and 40% of the examples are incompatible, respectively. This corresponds to a total of 15 points in the `correlation - incompatibility` space; as we already mentioned, for each such point we generate four random problems, for a total of 60 problems (each problem consists of 800 examples). [8]

ACKNOWLEDGMENTS

The authors are grateful to Daniel Marcu, Kevin Knight and Yolanda Gil their useful comments. The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory under contract/agreement numbers F30602-01-C-0197, F30602-00-1-0504, F30602-98-2-0109, in part by the Air Force Office of Scientific Research under grant number F49620-01-1-0053, in part by the National Science Foundation under award number DMI-0090978, and in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, cooperative agreement number EEC-9529152. The U.S.Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copy right annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

[8] The documents are tokenized, the UseNet headers are discarded, words on a stoplist are removed, no stemming is performed, and words that appear only in a single document are removed. The resulting views **V1** and **V2** have 5061 and 5385 features (i.e., words), respectively.