Selective Sampling + Semi-supervised Learning = Robust Multi-View Learning

Ion Muslea, Steven Minton, Craig A. Knoblock
muslea@isi.edu, minton@fetch.com, knoblock@isi.edu

Abstract

Although semi-supervised, multi-view algorithms have been successfully used to reduce the need for labeled data, our experiments indicate that existing approaches are sensitive to the levels of correlation and incompatibility of the views in a domain. To cope with this problem, we introduce a new multi-view algorithm, Co-Test(Co-EM), which interleaves active learning (Co-Testing) and semi-supervised learning (Co-EM). Co-Test(Co-EM) has a robust behavior on a large spectrum of problems because of its ability to ask for the labels of the most ambiguous examples, which compensates for the weaknesses of the underlying semi-supervised algorithm. We compare Co-Test(Co-EM) with four semi-supervised algorithms on a parameterized family of 60 text categorization problems in which we control the level of view correlation and incompatibility. The experiments show that Co-Test(Co-EM) clearly outperforms the other algorithms *without* using more labeled data; these results are further confirmed by two additional experiments performed on real-world datasets.

1 Introduction

Recently, an increasing number of machine learning researchers focused on problems with *multiple, redundant views*. In such problems, the domain's features can be split into several subsets, each of them being *sufficient* to learn the target concept. For instance, as described in [Blum and Mitchell, 1998], one can classify Web pages *either* based on the words within the documents *or* based on the words that appear in the hyperlinks pointing to them; or one can classify segments of televised broadcast based either on the video or on the audio information; or a robot can navigate based on either sonar or visual sensors. *Multi-view algorithms* have been mostly used to reduce the need for labeled data, and they have been successfully applied to a variety of real-world domains, from natural language processing [Riloff and Jones, 1999] [Collins and Singer, 1999] and speech recognition [de Sa and Ballard, 1998] to information extraction [Muslea *et al.*, 2000].

In this paper we first show that there are two crucial factors that influence the applicability of various multi-view algorithms: view *incompatibility* and view *correlation*. Intuitively, the former is a measure of how many examples are labeled differently by the target concepts in the different views. The latter is a measure of how much the values of the features in one view influence the values of the features in the other views. Based on a parameterized family of text classification problems in which we control both factors, we provide empirical evidence that in various regions of the `correlation - incompatibility` space, each of the three semi-supervised, multi-view algorithms considered here (Co-Training, Co-EM, and 2v-EM) is outperformed by some of the others.

To cope with this problem, we propose a novel multi-view algorithm called Co-Test(Co-EM), which combines an active learning algorithm (Co-Testing) with a semi-supervised one (Co-EM). On the regions of the `correlation - incompatibility` space where Co-EM outperforms the other algorithms, Co-Test(Co-EM) benefits from Co-EM's strength; on many of the other regions, Co-Testing's ability to make informative queries compensates for Co-EM's weaknesses. The final result is a robust algorithm that outperforms the semi-supervised algorithms on most regions in the `correlation - incompatibility` space.

This paper is structured as follows: after introducing the multi-view setting, we briefly present the semi-supervised algorithms used in the comparison. Then we motivate the need for a new, robust multi-view algorithm by providing empirical evidence on how the various algorithms outperform each other in various points in the `correlation - incompatibility` space. Finally, we introduce Co-Test(Co-EM), discuss the empirical results, and present directions for future work.

2 Issues in the Multi-View Setting

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

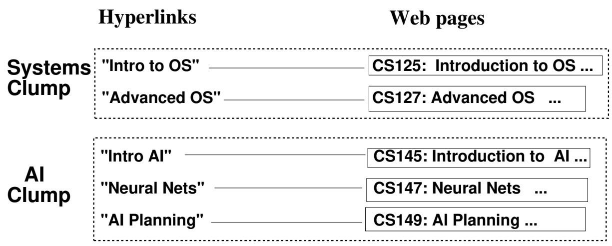

Figure 1: Two clumps in the "course homepage" class.

Given:
- a learning algorithm $\mathcal{L}$
- the sets T and U of labeled & unlabeled examples

- 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)
- $h = \mathcal{L}_{MAP}(New)$

Figure 2: **The EM Algorithm.**

Blum and Mitchell (1998) proved that given a few labeled examples and sufficiently many unlabeled examples, two *independent*, *compatible* views are sufficient to learn the target concept. The two key factors in their proof are the compatibility and independence of the views. The former requires that for *most* examples, the target concepts over each view predict the same label. The latter means that for any example $x = [x_1, x_2]$, x_1 and x_2 are conditionally independent given their label.[1]

The proof in [Blum and 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 independent, 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 independence and the compatibility of the views are crucial in this process:[2] if the views are not independent, the training set in **V2** is *not* random. Alternatively, 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 "miss-label" so many examples that learning the target concept in **V2** becomes impossible.

To understand how view compatibility and independence are violated in practice, let us consider the COURSES problem [Blum and Mitchell, 1998] in which Web pages must be categorized in two classes: "course homepages" and "other pages." For this problem, one can classify Web pages either based on the words in the page (view **V1**), or based on the words in the hyperlink that points to it (view **V2**). That is, the features in **V1** and **V2** consist of all the words that appear in Web pages and hyperlinks, respectively.

In most real world problems, the views are partially incompatible because of various reasons: corrupted features, insufficient attributes, etc. For instance, assume that a hyperlink that contains the text `''Neural Nets''` points to a neural nets class, while another hyperlink that contains *the same phrase* points to a publications page. That is, two Web pages with different labels in **V1** have the *same* description in **V2**. Consequently, either `[''Neural Networks'', ClassPage]` or `[''Neural Networks'', PubsPage]` are incompatible because the `''Neural Networks''` hyperlink can not have two different labels.

In many problems, the views are also (partially) correlated because of *domain clumpiness*, which can be best introduced by an example. For instance, hyperlinks that contain the words `''operating systems''` are more likely to point to systems courses than to AI courses. We say that the hyperlinks and homepages of the systems courses form a *clump* because they are highly likely to be "tied together" by an example (similarly, hyperlinks and homepages of AI classes form another clump; see Figure 1, which shows an extreme situation, with no inter-clump "connections"). Note that having two clumps per class is *sufficient* to violate the views' conditional independence because for any example $[x_1, x_2]$, it is highly likely that x_1 and x_2 come from the same clump.

[1] That is, the conditional probability that x has the description x_1 in **V1** *given* its description x_2 in **V2** *is the same as* the conditional probability of x having the description x_1 in **V1** *given* the label of x in **V2** (a similar condition must also hold for the description x_2 of x in **V2**).

[2] In an updated version of [Blum and Mitchell, 1998], the authors showed that the theoretical guarantees hold even for domains with a certain amount of incompatibility, provided that the views are independent. However, as in practice one rarely, if ever, encounters problems with independent views, one cannot ignore the importance of view incompatibility in real-world problem.

3 Semi-supervised Algorithms

In this section we provide a high-level description of the semi-supervised algorithms that are used in our comparison: EM, Co-Training, Co-EM, and 2v-EM.

3.1 The EM Algorithm

Semi-supervised EM [Nigam and Ghani, 2000], which is a single-view algorithm based on which Co-EM and 2v-EM were developed, applies a probabilistic learning algorithm $\mathcal{L}$ to a small set of labeled examples and a large set of unlabeled ones (see Figure 2). First, EM creates an initial classifier h based solely on the labeled examples. Then it repeatedly performs a two-step procedure:

- use h to probabilistically label all unlabeled examples;
- 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.

3.2 Co-Training

Co-Training [Blum and Mitchell, 1998] is a semi-supervised algorithm that uses the initial training set to learn a (weak)

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

`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

Figure 3: **The Co-Training Algorithm.**

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 3). 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.

3.3 Co-EM

As shown in Figure 4, Co-EM [Nigam and Ghani, 2000] uses the hypothesis learned in one view to probabilistically label the examples in the other view. More precisely, during each iteration, we learn one hypothesis in each view; then we use each hypothesis to label all the unlabeled examples, which will be then used to train *the other view* during the next iteration.

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. By contrast, Co-Training's commitment to the highly confident predictions may add to the training set a large number of mislabeled examples, especially during the first iterations, when the weak hypotheses may have little prediction power.[3]

Intuitively, Co-EM decouples the two sources of "noise" that may force EM into a local minimum: the set of features

[3]An alternative description of the difference between Co-EM and Co-Training is provided in [Nigam and Ghani, 2000], where the authors characterize the two algorithms as *iterative* and *incremental*, respectivelly. The two descriptions are equivalent: Co-EM *iteratively* uses the unlabeled data, which means that it *does not commit* to the labels learned during the previous iteration. By contrast, Co-Training *incrementally* uses the unlabeled data by *committing* only to a few labels during each iteration and ignoring the remaining unlabeled examples.

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

- let *All* be the unlabeled set of all examples from T and U;
- $h_1 = \mathcal{L}_{MAP}(\mathbf{V1}(T))$

`LOOP` for k iterations
- New_1 = `ProbabilisticallyLabel`(*All*, h_1)
- $h_2 = \mathcal{L}_{MAP}(\mathbf{V2}(New_1))$

- New_2 = `ProbabilisticallyLabel`(*All*, h_2)
- $h_1 = \mathcal{L}_{MAP}(\mathbf{V1}(New_2))$

- combine the prediction of h_1 and h_2

Figure 4: **The Co-EM Algorithm.**

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

- create classifier h_1 by performing EM on **V1**
- create classifier h_2 by performing EM on **V2**
- combine the prediction of h_1 and h_2

Figure 5: **The 2v-EM Algorithm.**

and the learning algorithm. If the two views are independent, the noisy labels produced by one view are independent of the imperfect features used in the learning process within the other view. By contrast, in a single view framework, the two sources of noise may amplify each other, which may lead to EM getting stuck easily in a local minimum.

3.4 2v-EM

We created 2v-EM just to show that the strength of Co-Training and Co-EM does not come merely from combining classifiers learned in different views, but rather from using knowledge acquired in one view to train the other one. The 2v-EM algorithm is a straightforward extension of EM for problems with multiple views: it first performs EM on each view in isolation, and then it combines the prediction of the hypotheses learned in each view (see Figure 5). It turns out that as the degree of view incompatibility increases, 2v-EM outperforms the other algorithms because the knowledge acquired in one view becomes misleading in the other one.

3.5 An Empirical Comparison

In practice, one rarely, if ever, finds problems with totally compatible, fully independent views. Consequently, it is crucial to understand how robust multi-view algorithms are with respect to these two factors. We performed an empirical comparison of the four semi-supervised algorithms on a parameterized family of 60 *semi-artificially* created problems (i.e., they are created based on a text classification domain, on top of which we added various levels of view correlation and incompatibility; see Appendix `A` for details).

To create the 60 problems, we consider 15 points in the `correlation - incompatibility` space; for each such point, we generate four random problems based on the

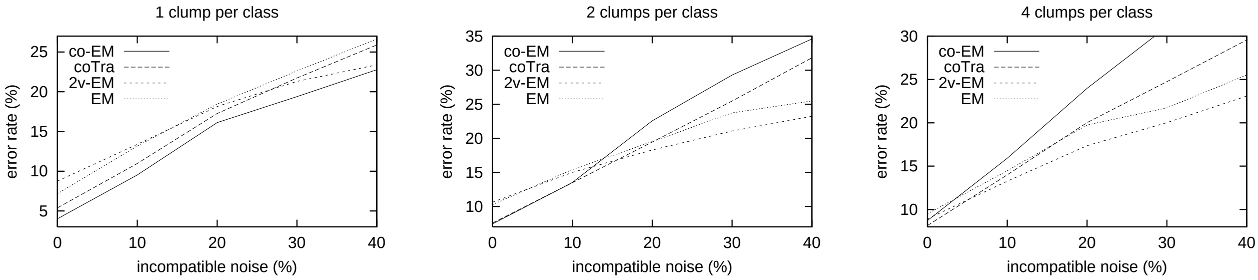

Figure 6: A comparison of four semi-supervised algorithms.

examples in the original domain. Each point in the `correlation - incompatibility` space is characterized by two parameters: clumpiness (one, two, and four clumps per class), and incompatibility (0%, 10%, 20%, 30%, and 40% of the examples are incompatible).

Figure 6 shows the performance of the four semi-supervised algorithms on the 60 problems.[4] Each graph displays the results in five points of the `correlation - incompatibility` space; these five points have the same degree of view correlation, but represent problems with different degrees of view incompatibility. Within each graph, on the `x` axis we have the percentage of incompatible examples in each problem. The `y` axis represents error rates, which are averaged over the four random problems that we generated for each point of the `correlation - incompatibility` space. Within Figure 6, the degree of view correlation increases from left to right: in the left-most graph, problems have conditionally independent views, while in the middle and the right-most ones, problems have two and four clumps per class, respectively.

From these experiments we can draw three interesting observations. First, on problems with conditionally independent views, Co-EM outperforms the other three algorithms. Second, on clumpy problems with 20% or more incompatible examples, 2v-EM outperforms all other algorithms. Finally, in the remaining problems no algorithm is clearly better than the other ones.

The results above show that each algorithm is sensitive to variations of the two key factors: view incompatibility and correlation. To address this problem, we propose a novel, robust algorithm, Co-Test(Co-EM), which is described in the next section.

4 Co-Testing + Co-EM = Co-Test(Co-EM)

Co-Testing [Muslea *et al.*, 2000] is a family of active learning algorithms that was designed for problems with multiple views. Co-Testing starts with a few labeled examples and a pool of unlabeled ones (see Figure 7) and assumes that the user is willing to label a few more examples. Consequently, Co-Testing searches for the most informative examples in the unlabeled pool and asks the user to label them. Co-Testing repeatedly trains one hypothesis for each view and selects as a query one of the unlabeled examples on which

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

LOOP for k iterations
- use $\mathcal{L}$, **V1**(T), and **V2**(T) to create classifiers h_1 and h_2
- let $ContentionPoints$ = $\{ x \in U, h_1(x) \neq h_2(x) \}$
- let x = **SelectQuery**($ContentionPoints$)
- remove x from U, ask the user to label it, and add it to T

- combine the prediction of h_1 and h_2

Figure 7: **The Co-Testing Family of Algorithms.**

the two hypotheses predict different labels (also called *contention points*). Intuitively, if two *compatible* views do not agree about a label, at least one of them must be wrong. Consequently, by asking for the label of a contention point x, Co-Testing is likely to provide useful information for the view that mislabeled x.

Co-Test(Co-EM) is a novel algorithm that interleaves Co-EM and Co-Testing. The only modification to the algorithm described in Figure 7 is that instead of learning h_1 and h_2 based only on the labeled examples, we obtain h_1 and h_2 by running Co-EM on both labeled and unlabeled examples. As we expect the hypotheses generated by Co-EM to be more accurate than the ones created solely based on the labeled data, it follows that Co-Test(Co-EM) should converge faster than Co-Testing.

Co-Test(Co-EM) is not the first algorithm that combines semi-supervised and active learning: in [McCallum and Nigam, 1998], the authors show that various combinations of EM and Query-by-Committee (QBC) perform better than the stand-alone algorithms. McCallum and Nigam also reported that, surprisingly, in their (single-view) approach, interleaving QBC and EM did not perform as well as EM applied *after* all the queries are made. By contrast, in our preliminary experiments, we noticed that if we do not apply Co-EM after each query, the error rate increases significantly.

The various members of the Co-Testing family differ from each other by the manner in which they select the next query. Co-Test(Co-EM) uses a straightforward strategy to pick the next query: among the contention points, it selects the one on which the combined prediction of h_1 and h_2 is the least confident.

[4]For each problem, we used two random runs of 5-fold cross-validation. The four semi-supervised algorithms used 40 labeled and 600 unlabeled examples; the test sets contained 160 examples.

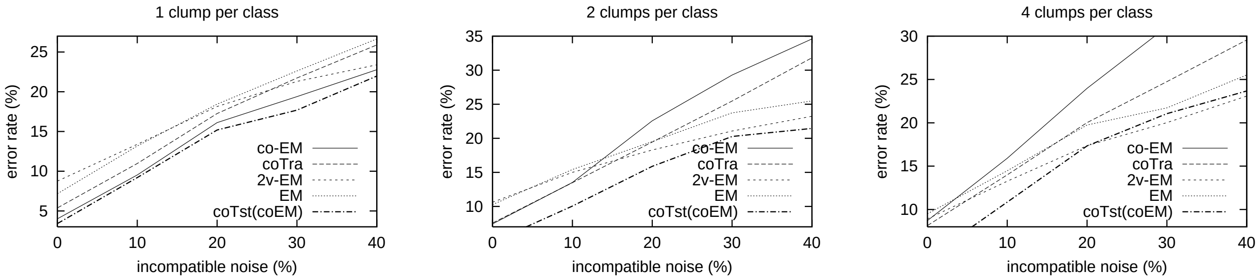

Figure 8: Co-Test(Co-EM) *vs* semi-supervised algorithms.

5 Empirical Results

We applied EM, 2v-EM, Co-Training, Co-EM, and Co-Test(Co-EM) to the family of 60 problems described in Appendix **A** (for each problem we perform two random runs of 5-fold cross-validation). The semi-supervised algorithms use a training set of 40 labeled examples and a pool of 600 unlabeled examples, while the test set consists of 160 examples. To keep the comparison fair, Co-Test(Co-EM) starts with 10 random examples and selects 30 queries among the 600 unlabeled examples, for a total of 40 labeled examples.

All five algorithms in our comparison use Naive Bayes as the underlying algorithm $\mathcal{L}$. For EM, Co-Training, Co-EM, and Naive Bayes, we implemented the versions described in [Nigam and Ghani, 2000]. EM, 2v-EM, and Co-EM were run to convergence, which required seven, seven, and five iterations, respectively. Co-Training required a fine tuning of both the number of iteration and the number of examples that are added to the training set after each iteration; eventually, we set these parameters to seven and 40, respectively. To avoid prohibitive running time, within Co-Test(Co-EM) we ran Co-EM only for two iterations (on each of the 60 problems, Co-Test(Co-EM) runs Co-EM after each of the 300 queries: 2 runs $\times$ 5 folds $\times$ 30 queries per fold). For all algorithms, we consider the error rate obtained after the last iteration; at each point in the `correlation - incompatibility` space, the reported error rate is averaged over four random problems.

As shown in Figure 8,[5] Co-Test(Co-EM) has the best results on most of the problems: it obtains the smallest error rates on 13 of the 15 points that we considered in the `correlation - incompatibility` space. On the two remaining points, which correspond to the most clumpy, incompatible views, Co-Test(Co-EM) loses to 2v-EM, but outperforms the other algorithms. As the view incompatibility rises, Co-Test(Co-EM)'s performance is (negatively) influenced by an inherent weakness of algorithms like Co-EM and Co-Training: for many examples, the labels learned in one view are incorrect in the other one. By contrast, 2v-EM, which learns the hypotheses independently of each other, does not have to cope with this problem. In final analysis, Co-Test(Co-EM) clearly outperforms the other algorithms on all problems in which the views are sufficiently compatible to allow useful information exchange between the views.

The most spectacular improvements of Co-Test(Co-EM) over Co-EM are on the clumpy domains, where it outperforms Co-EM, Co-Training, and EM on all 10 points! Intuitively, Co-EM has a poor performance on clumpy domains because the information exchanged between views remains localized within each clump. Consequently, one way to improve Co-EM is to accurately detect the boundaries between classes, thus minimizing the amount of mislabeled examples exchanged between views. We believe that by using Co-Testing to detect the most ambiguous contention points, Co-Test(Co-EM) provides Co-EM a better "description" of this decision boundary then the one obtained based on random examples.

On problems with highly compatible, conditionally independent views (leftmost points in the leftmost graph), even though Co-Test(Co-EM) and Co-EM outperform the other algorithms, there is no clear winner between the two of them. As Co-Test(Co-EM) uses Co-Testing mainly to help Co-EM cope with domain clumpiness, this result was to be expected. A complementary explanation for Co-Test(Co-EM)'s inability to outperform Co-EM more clearly on such problems is closely related to the theoretical argument discussed in Section 2: on problems with independent, compatible views a few labeled examples are sufficient to learn the target concept. As the views become less compatible, the information learned in one view becomes misleading in the other view, and Co-Testing cannot help here.

From a practical perspective, we conjecture that Co-Test(Co-EM) has the best potential of all multi-view algorithms discussed here. We believe that most real-world, multi-view problems display some (low) level of view incompatibility and correlation, which is the spectrum of problems where Co-Test(Co-EM) performs the best. A preliminary test confirms our conjecture: we compared the five algorithms above on the COURSES and POLITICS domains,[6] and

[5] In Figure 8, the results for the four passive semi-supervised algorithm are the ones that we have shown in Figure 6.

[6] In the POLITICS domain, which contains the postings in the `talk.politics.guns` and `talk.politics.misc` newsgroups [Joachims, 1996], the system must determine the newsgroup in which each article was posted. The two views consists of the words and 2-grams (i.e., pairs of two consecutive words) that appear in the documents, respectively. When tokenizing the documents, we discard words that appear in a stop list, in the UseNet headers, or in less then 10 documents. Similarly, in the COURSES domain [Blum and Mitchell, 1998], which was described in Section 2, we remove the HTML code and ignore words that appear in less than 10 documents.

To evaluate the error rates, for both domains we used two random runs of 5-fold cross validation. On COURSES (1041 examples) and

Co-Test(Co-EM) obtained again the smallest error rates. The results in Table 1 (the error rates for COURSES and POLITICS are on the first and second line, respectively) confirm our earlier conclusions. First, Co-Test(Co-EM) outperforms the semi-supervised algorithms. Second, the relative performance of the various semi-supervised algorithms varies drastically from problem to problem.

6 Conclusions and Future Work

We analyzed two main factors that influence the applicability of multi-view algorithms: view independence and compatibility. We showed that even though semi-supervised, multi-view algorithms tend to outperform single-view EM, in various regions of the `correlation - incompatibility` space we have different winners. To cope with this problem, we proposed a robust multi-view algorithm that outperforms the semi-supervised algorithms on most problems.

We plan to continue our work along two main directions. First, we will continue to study various combinations of Co-Testing and semi-supervised algorithms, both on semi-artificial and real-world domains. Second, we plan to search for a formal way to detect the redundant views within a given domain.

References

[Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of Computational Learning Theory*, pages 92–100, 1998.

[Collins and Singer, 1999] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Conf. Very Large Corpora VLC-99*, pages 100–110, 1999.

[de Sa and Ballard, 1998] V. de Sa and D. Ballard. Category learning through multi-modality sensing. *Neural Computation*, 10(5):1097–1117, 1998.

[Joachims, 1996] T. Joachims. A probabilistic analysis of the ROCCHIO algorithm with TFIDF for text categorization. In *Computer Sci. Tech. Report CMU-CS-96-118*, 1996.

[McCallum and Nigam, 1998] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 359–367, 1998.

[Muslea *et al.*, 2000] I. Muslea, S. Minton, and C. Knoblock. Selective sampling with redundant views. In *Proc. of AAAI-2000*, pages 621–626, 2000.

POLITICS (2000 examples), the semi-supervised algorithms use 40 and 100 labeled examples, respectively. On COURSES, Co-Test(Co-EM) starts with 6 labeled examples and makes 34 queries; on POLITICS, it starts with 10 labeled examples and makes 90 queries. To reach convergence, EM and 2v-EM run for 7 iterations on both domains; Co-EM, both stand-alone and within Co-Test(Co-EM), runs for two iterations on COURSES and three iterations on POLITICS. For COURSE, Co-Training labels 50 examples after each of the 7 iteration; for POLITICS, it labels 100 examples after each of the 8 iterations.

coTstEM	co-EM	EM	coTrain	2v-EM
3.07 ± 0.66	3.93 ± 0.48	5.23 ± 0.86	5.85 ± 0.60	6.47 ± 1.15
15.85 ± 1.55	17.72 ± 1.29	17.40 ± 1.41	16.47 ± 1.38	17.70 ± 1.32

Table 1: **Error rates on** COURSES **and** POLITICS**.**

[Nigam and Ghani, 2000] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of CIKM-2000*, 2000.

[Riloff and Jones, 1999] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proc. of the AAAI-1999*, pages 474–479, 1999.

A The 60 Semi-Artificial Problems

To create a parameterized family of problems in which we can manipulate the two parameters of interest, we extend an idea from [Nigam and Ghani, 2000]: one can create a (semi-artificial) domain with two independent views by taking two *unrelated* problems with the same number of positive and negative examples and considering each problem an individual view. By randomly pairing together positive examples from **V1** and **V2**, and negative examples from **V1** and **V2**, one obtains a problem with two compatible views that are conditionally independent given the label.

This procedure can be easily modified to also create incompatible examples. First, we randomly pick one positive and one negative example, say `[''Advanced OS'', AdvOS-Class]` and `[''John Doe'', JohnDoe-Homepage]`. Second, we replace these two examples by their "recombinations", `[''Advanced OS'', JohnDoe-Homepage]` and `[''Joe Doe'', AdvOS-Class]`, which are positive in one view and negative in the other one.

To simulate various degrees of clumpiness, we convert a classification problem with 16 equally-populated classes into two binary classification problems; the two problems then become the two views, while each of the 16 classes represents a (potential) clump. For this task we used the Mini-Newsgroups dataset, which is a subset of the well-known `20-Newsgroups` dataset [Joachims, 1996].[7] The domain consists of 100 articles from 20 Usenet newsgroups; the 100 examples were randomly chosen from the 1000 postings included in the original `20-Newsgroups` dataset.

We selected 16 of the 20 newsgroups, and we divided them in four groups of four (see Table 2). The examples in each such group were used as either positive or negative examples in one of the two views. We generated three levels of clumpiness: one, two, and four clumps per class, respectively. Figure 9 shows how the three levels of clumpiness are created for the positive examples (the same methodology was also used for the negative examples). For one

[7] As our empirical study compares 5 algorithms over 60 "instances" of the dataset, using the entire 20 Newsgroups dataset would have been prohibitive in terms of running time. The Mini-Newsgroups are available at http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/mini_newsgroups.tar.gz

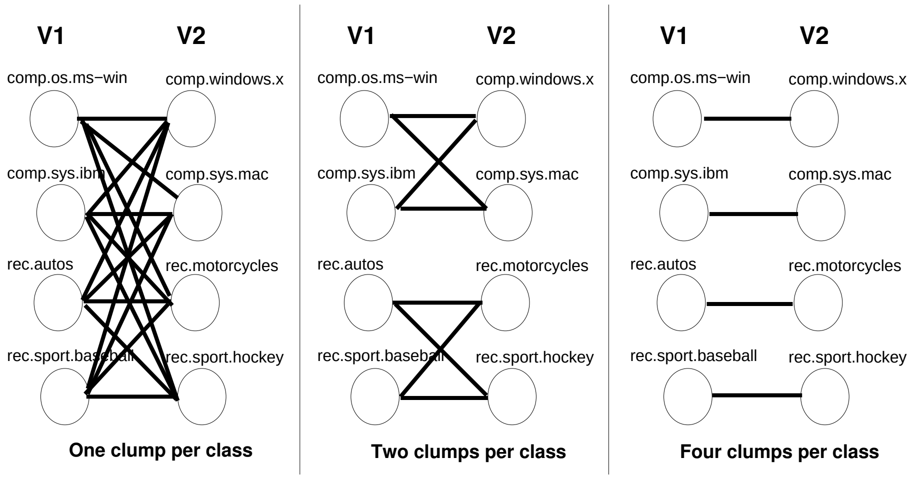

Figure 9: **Generating the clumps.**

	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.**

clump per class, any positive example from **V1** can be "connected" to any positive example in **V2** (this corresponds to conditionally independent views). For two clumps per class, we do *not* allow connections between the `comp` examples in one view and the `rec` examples in the other one. Finally, for four clumps per class we randomly connect examples from `comp.os.ms-win` to `comp.windows.x`, from `comp.sys.ibm` to `comp.sys.mac`, etc.[8]

Each of these three levels of clumpiness is considered with five degrees of view incompatibility: 0%, 10%, 20%, 30%, and 40% of the examples are corrupted, 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. The documents that represent the components of each example in the two views 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. **V1** and **V2** have 5061 and 5385 features (i.e., words), respectively.

[8]In Figure 9, the line that connects a newsgroups A in **V1** to a newsgroup B in **V2** denotes that the domain includes examples [a, b], where $a \in A$ and $b \in B$. Consequently, at one clump per class, each newsgroup is connected to *all* newsgroups in the other view; by contrast, at four clumps per class, each newsgroup is connect to a *single* newsgroup in the other view.

Acknowledgments

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under contract number F30602-99-1-0524, in part by the Rome Laboratory of the Air Force Systems Command and the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-98-2-0109, in part by the United States Air Force under contract number F49620-98-1-0046, and in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policy or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.