Surrogate Learning - From Feature Independence to Semi-Supervised Classification

Sriharsha Veeramachaneni and **Ravi Kumar Kondadadi**
Thomson Reuters Research and Development
Eagan, MN 55123, USA
[harsha.veeramachaneni,ravikumar.kondadadi]@thomsonreuters.com

Abstract

We consider the task of learning a classifier from the feature space $\mathcal{X}$ to the set of classes $\mathcal{Y} = \{0, 1\}$, when the features can be partitioned into class-conditionally independent feature sets $\mathcal{X}_1$ and $\mathcal{X}_2$. We show that the class-conditional independence can be used to represent the original learning task in terms of 1) learning a classifier from $\mathcal{X}_2$ to $\mathcal{X}_1$ (in the sense of estimating the probability $P(\mathbf{x}_1|\mathbf{x}_2)$)and 2) learning the class-conditional distribution of the feature set $\mathcal{X}_1$. This fact can be exploited for semi-supervised learning because the former task can be accomplished purely from unlabeled samples. We present experimental evaluation of the idea in two real world applications.

1 Introduction

Semi-supervised learning is said to occur when the learner exploits (a presumably large quantity of) unlabeled data to supplement a relatively small labeled sample, for accurate induction. The high cost of labeled data and the simultaneous plenitude of unlabeled data in many application domains, has led to considerable interest in semi-supervised learning in recent years (Chapelle et al., 2006).

We show a somewhat surprising consequence of class-conditional feature independence that leads to a principled and easily implementable semi-supervised learning algorithm. When the feature set can be partitioned into two class-conditionally independent sets, we show that the original learning problem can be reformulated in terms of the problem of learning a first predictor from one of the partitions to the other, plus a second predictor from the latter partition to class label. That is, the latter partition acts as a *surrogate* for the class variable. Assuming that the second predictor can be learned from a relatively small labeled sample this results in an effective semi-supervised algorithm, since the first predictor can be learned from only unlabeled samples.

In the next section we present the simple yet interesting result on which our semi-supervised learning algorithm (which we call *surrogate learning*) is based. We present examples to clarify the intuition behind the approach and present a special case of our approach that is used in the applications section. We then examine related ideas in previous work and situate our algorithm among previous approaches to semi-supervised learning. We present empirical evaluation on two real world applications where the required assumptions of our algorithm are satisfied.

2 Surrogate Learning

We consider the problem of learning a classifier from the feature space $\mathcal{X}$ to the set of classes $\mathcal{Y} = \{0, 1\}$. Let the features be partitioned into $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$. The random feature vector $\mathbf{x} \in \mathcal{X}$ will be represented correspondingly as $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$. Since we restrict our consideration to a two-class problem, the construction of the classifier involves the estimation of the probability $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$ at every point $(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}$.

We make the following assumptions on the joint probabilities of the classes and features.

Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing, pages 10–18,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

1. $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{y}) = P(\mathbf{x}_1|\mathbf{y})P(\mathbf{x}_2|\mathbf{y})$ for $\mathbf{y} \in \{0, 1\}$. That is, the feature sets $\mathbf{x}_1$ and $\mathbf{x}_2$ are class-conditionally independent for both classes. Note that, when $\mathcal{X}_1$ and $\mathcal{X}_2$ are one-dimensional, this condition is identical to the *Naive Bayes* assumption, although in general our assumption is weaker.

2. $P(\mathbf{x}_1|\mathbf{x}_2) \neq 0$, $P(\mathbf{x}_1|\mathbf{y}) \neq 0$ and $P(\mathbf{x}_1|\mathbf{y} = 0) \neq P(\mathbf{x}_1|\mathbf{y} = 1)$. These assumptions are to avoid *divide-by-zero* problems in the algebra below. If $\mathbf{x}_1$ is a discrete valued random variable and not irrelevant for the classification task, these conditions are often satisfied.

We can now show that $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$ can be written as a function of $P(\mathbf{x}_1|\mathbf{x}_2)$ and $P(\mathbf{x}_1|\mathbf{y})$. When we consider the quantity $P(\mathbf{y}, \mathbf{x}_1|\mathbf{x}_2)$, we may derive the following.

$$\begin{aligned}
& P(\mathbf{y}, \mathbf{x}_1|\mathbf{x}_2) = P(\mathbf{x}_1|\mathbf{y}, \mathbf{x}_2)P(\mathbf{y}|\mathbf{x}_2) \\
\Rightarrow\ & P(\mathbf{y}, \mathbf{x}_1|\mathbf{x}_2) = P(\mathbf{x}_1|\mathbf{y})P(\mathbf{y}|\mathbf{x}_2) \\
& \quad \text{(from the independence assumption)} \\
\Rightarrow\ & P(\mathbf{y}|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_1|\mathbf{x}_2) = P(\mathbf{x}_1|\mathbf{y})P(\mathbf{y}|\mathbf{x}_2) \\
\Rightarrow\ & \frac{P(\mathbf{y}|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y})} = P(\mathbf{y}|\mathbf{x}_2) \qquad (1)
\end{aligned}$$

Since $P(\mathbf{y} = 0|\mathbf{x}_2) + P(\mathbf{y} = 1|\mathbf{x}_2) = 1$, Equation 1 implies

$$\begin{aligned}
& \frac{P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y} = 0)} + \\
& \qquad \frac{P(\mathbf{y} = 1|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y} = 1)} = 1 \\
\Rightarrow\ & \frac{P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y} = 0)} + \\
& \qquad \frac{(1 - P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2))\, P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y} = 1)} = 1 (2)
\end{aligned}$$

Solving Equation 2 for $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$, we obtain

$$\begin{aligned}
& P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2) = \\
& \quad \frac{P(\mathbf{x}_1|\mathbf{y} = 0)}{P(\mathbf{x}_1|\mathbf{x}_2)} \cdot \frac{P(\mathbf{x}_1|\mathbf{y} = 1) - P(\mathbf{x}_1|\mathbf{x}_2)}{P(\mathbf{x}_1|\mathbf{y} = 1) - P(\mathbf{x}_1|\mathbf{y} = 0)} (3)
\end{aligned}$$

We have succeeded in writing $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$ as a function of $P(\mathbf{x}_1|\mathbf{x}_2)$ and $P(\mathbf{x}_1|\mathbf{y})$. Although this result was previously observed in a different context by Abney in (Abney, 2002), he does not use it to derive a semi-supervised learning algorithm. This result can lead to a significant simplification of the learning task when a large amount of unlabeled data is available. The semi-supervised learning algorithm involves the following two steps.

1. From unlabeled data learn a predictor from the feature space $\mathcal{X}_2$ to the space $\mathcal{X}_1$ to predict $P(\mathbf{x}_1|\mathbf{x}_2)$. There is no restriction on the learner that can be used as long as it outputs posterior class probability estimates.

2. Estimate the quantity $P(\mathbf{x}_1|\mathbf{y})$ from a labeled samples. In case $\mathbf{x}_1$ is finite valued, this can be done by just counting. If $\mathcal{X}_1$ has low cardinality the estimation problem requires very few labeled samples. For example, if $\mathbf{x}_1$ is binary, then estimating $P(\mathbf{x}_1|\mathbf{y})$ involves estimating just two Bernoulli probabilities.

Thus, we can decouple the prediction problem into two separate tasks, one of which involves predicting $\mathbf{x}_1$ from the remaining features. In other words, $\mathbf{x}_1$ serves as a *surrogate* for the class label. Furthermore, for the two steps above there is no necessity for complete samples. The labeled examples can have the feature $\mathbf{x}_2$ missing.

At test time, an input sample $(\mathbf{x}_1, \mathbf{x}_2)$ is classified by computing $P(\mathbf{x}_1|\mathbf{y})$ and $P(\mathbf{x}_1|\mathbf{x}_2)$ from the predictors obtained from training, and plugging these values into Equation 3. Note that these two quantities are computed for the actual value of $\mathbf{x}_1$ taken by the input sample.

The following example illustrates surrogate learning.

Example 1
Consider the following variation on a problem from (Duda et al., 2000) of classifying fish on a converyor belt as either *salmon* ($\mathbf{y} = 0$) or *sea bass* ($\mathbf{y} = 1$). The features describing the fish are $\mathbf{x}_1$, a binary feature describing whether the fish is *light* ($\mathbf{x}_1 = 0$) or *dark* ($\mathbf{x}_1 = 1$), and $\mathbf{x}_2$ describes the length of the fish which is real-valued. Assume (unrealistically) that $P(\mathbf{x}_2|\mathbf{y})$, the class-conditional distribution of $\mathbf{x}_2$, the length for *salmon* is Gaussian,

and for the *sea bass* is Laplacian as shown in Figure 1.

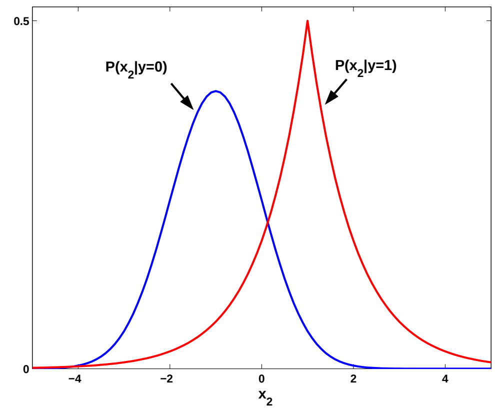

Figure 1: Class-conditional probability distributions of the feature $\mathbf{x}_2$.

Because of the class-conditional feature independence assumption, the joint distribution $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = P(\mathbf{x}_2|\mathbf{y})P(\mathbf{x}_1, \mathbf{y})$ can now be completely specified by fixing the joint probability $P(\mathbf{x}_1, \mathbf{y})$. Let $P(\mathbf{x}_1 = 0, \mathbf{y} = 0) = 0.3$, $P(\mathbf{x}_1 = 0, \mathbf{y} = 1) = 0.1$, $P(\mathbf{x}_1 = 1, \mathbf{y} = 0) = 0.2$, and $P(\mathbf{x}_1 = 1, \mathbf{y} = 1) = 0.4$. I.e., a *salmon* is more likely to be *light* than *dark* and a *sea bass* is more likely to be *dark* than *light*.

The full joint distribution is depicted in Figure 2. Also shown in Figure 2 are the conditional distributions $P(\mathbf{x}_1 = 0|\mathbf{x}_2)$ and $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$.

Assume that we build a predictor to decide between $\mathbf{x}_1 =$ *light* and $\mathbf{x}_1 =$ *dark* from the *length* using a data set of unlabeled fish. On a random *salmon*, this predictor will most likely decide that $\mathbf{x}_1 =$ *light* (because, for a *salmon*, $\mathbf{x}_1 =$ *light* is more likely than $\mathbf{x}_1 =$ *dark*, and similarly for a *sea bass* the predictor often decides that $\mathbf{x}_1 =$ *dark*. Consequently the predictor provides information about the true class label $\mathbf{y}$. This can also be seen in the similarities between the curves $P(\mathbf{y} = 0|\mathbf{x}_1, \mathbf{x}_2)$ to the curve $P(\mathbf{x}_1|\mathbf{x}_2)$ in Figure 2.

Another way to interpret the example is to note that if a predictor for $P(\mathbf{x}_1|\mathbf{x}_2)$ were built on *only* the *salmons* then $P(\mathbf{x}_1 = \mathit{light}|\mathbf{x}_2)$ will be a constant value (0.6). Similarly the value of $P(\mathbf{x}_1 = \mathit{light}|\mathbf{x}_2)$ for *sea basses* will also be a constant value (0.2). That is, the value of $P(\mathbf{x}_1 = \mathit{light}|\mathbf{x}_2)$ for a sample is a good predictor of its class. However, surrogate learning builds the predictor $P(\mathbf{x}_1|\mathbf{x}_2)$ on unlabeled data from *both* types of fish and therefore additionally requires $P(\mathbf{x}_1|\mathbf{y})$ to estimate the boundary between the classes.

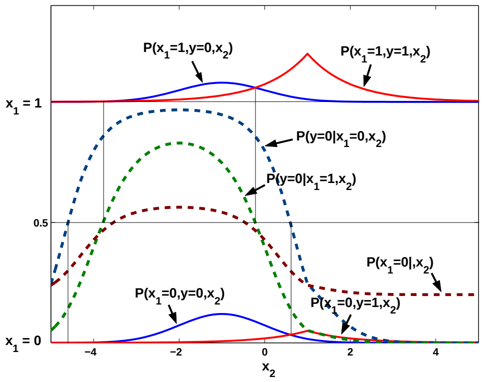

Figure 2: The joint distributions and the posterior distributions of the class $\mathbf{y}$ and the surrogate class $\mathbf{x}_1$.

2.1 A Special Case

The independence assumptions made in the setting above may seem too strong to hold in real problems, especially because the feature sets are required to be class-conditionally independent for *both* classes. We now specialize the setting of the classification problem to the one realized in the applications we present later.

We still wish to learn a classifier from $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ to the set of classes $\mathcal{Y} = \{0, 1\}$. We make the following slightly modified assumptions.

1. $\mathbf{x}_1$ is a binary random variable. That is, $\mathcal{X}_1 = \{0, 1\}$.

2. $P(\mathbf{x}_1, \mathbf{x}_2|\mathbf{y} = 0) = P(\mathbf{x}_1|\mathbf{y} = 0)P(\mathbf{x}_2|\mathbf{y} = 0)$. We require that the feature $\mathbf{x}_1$ be class-conditionally independent of the remaining features *only* for the class $\mathbf{y} = 0$.

3. $P(\mathbf{x}_1 = 0, \mathbf{y} = 1) = 0$. This assumption says that $\mathbf{x}_1$ is a '100% recall' feature for $\mathbf{y} = 1$[1].

Assumption 3 simplifies the learning task to the estimation of the probability $P(\mathbf{y} = 0|\mathbf{x}_1 = 1, \mathbf{x}_2)$ for every point $\mathbf{x}_2 \in \mathcal{X}_2$. We can proceed as before

[1] This assumption can be seen to trivially enforce the independence of the features for class $\mathbf{y} = 1$.

to obtain the expression in Equation 3.

$$\begin{aligned}
&P(\mathbf{y}=0|\mathbf{x}_1=1,\mathbf{x}_2)\\
&= \frac{P(\mathbf{x}_1=1|\mathbf{y}=0)}{P(\mathbf{x}_1=1|\mathbf{x}_2)}\cdots\\
&\quad\cdots\frac{P(\mathbf{x}_1=1|\mathbf{y}=1)-P(\mathbf{x}_1=1|\mathbf{x}_2)}{P(\mathbf{x}_1=1|\mathbf{y}=1)-P(\mathbf{x}_1=1|\mathbf{y}=0)}\\
&= \frac{P(\mathbf{x}_1=1|\mathbf{y}=0)}{P(\mathbf{x}_1=1|\mathbf{x}_2)}\cdot\frac{1-P(\mathbf{x}_1=1|\mathbf{x}_2)}{1-P(\mathbf{x}_1=1|\mathbf{y}=0)}\\
&= \frac{P(\mathbf{x}_1=1|\mathbf{y}=0)}{P(\mathbf{x}_1=1|\mathbf{x}_2)}\cdot\frac{P(\mathbf{x}_1=0|\mathbf{x}_2)}{P(\mathbf{x}_1=0|\mathbf{y}=0)}\\
&= \frac{P(\mathbf{x}_1=1|\mathbf{y}=0)}{P(\mathbf{x}_1=0|\mathbf{y}=0)}\cdot\frac{P(\mathbf{x}_1=0|\mathbf{x}_2)}{(1-P(\mathbf{x}_1=0|\mathbf{x}_2))} \quad (4)
\end{aligned}$$

Equation 4 shows that $P(\mathbf{y} = 0|\mathbf{x}_1 = 1, \mathbf{x}_2)$ is a monotonically increasing function of $P(\mathbf{x}_1 = 0|\mathbf{x}_2)$. This means that after we build a predictor from $\mathcal{X}_2$ to $\mathcal{X}_1$, we only need to establish the threshold on $P(\mathbf{x}_1 = 0|\mathbf{x}_2)$ to yield the optimum classification between $\mathbf{y} = 0$ and $\mathbf{y} = 1$. Therefore the learning proceeds as follows.

1. From unlabeled data learn a predictor from the feature space $\mathcal{X}_2$ to the binary space $\mathcal{X}_1$ to predict the quantity $P(\mathbf{x}_1|\mathbf{x}_2)$.

2. Use labeled sample to establish the threshold on $P(\mathbf{x}_1 = 0|\mathbf{x}_2)$ to achieve the desired precision-recall trade-off for the original classification problem.

Because of our assumptions, for a sample from class $\mathbf{y} = 0$ it is impossible to predict whether $\mathbf{x}_1 = 0$ or $\mathbf{x}_1 = 1$ better than random by looking at the $\mathbf{x}_2$ feature, whereas a sample from the positive class always has $\mathbf{x}_1 = 1$. Therefore the samples with $\mathbf{x}_1 = 0$ serve to delineate the positive examples among the samples with $\mathbf{x}_1 = 1$. We therefore call the samples that have $\mathbf{x}_1 = 1$ as the *target* samples and those that have $\mathbf{x}_1 = 0$ as the *background* samples.

3 Related Work

Although the idea of using unlabeled data to improve classifier accuracy has been around for several decades (Nagy and Shelton, 1966), semi-supervised learning has received much attention recently due to impressive results in some domains. The compilation of chapters edited by Chappelle et al. is an excellent introduction to the various approaches to semi-supervised learning, and the related practical and theoretical issues (Chapelle et al., 2006).

Similar to our setup, *co-training* assumes that the features can be split into two class-conditionally independent sets or 'views' (Blum and Mitchell, 1998). Also assumed is the sufficiency of either view for accurate classification. The co-training algorithm iteratively uses the unlabeled data classified with high confidence by the classifier on one view, to generate labeled data for learning the classifier on the other.

The intuition underlying co-training is that the errors caused by the classifier on one view are independent of the other view, hence can be conceived as uniform[2] noise added to the training examples for the other view. Consequently, the number of label errors in a region in the feature space is proportional to the number of samples in the region. If the former classifier is reasonably accurate, the *proportionally* distributed errors are 'washed out' by the correctly labeled examples for the latter classifier. Seeger showed that co-training can also be viewed as an instance of the Expectation-Maximization algorithm (Seeger, 2000).

The main distinction of surrogate learning from co-training is the learning of a predictor from one view to the other, as opposed to learning predictors from both views to the class label. We can therefore eliminate the requirement that both views be sufficiently informative for reasonably accurate prediction. Furthermore, unlike co-training, surrogate learning has no iterative component.

Ando and Zhang propose an algorithm to regularize the hypothesis space by simultaneously considering multiple classification tasks on the same feature space (Ando and Zhang, 2005). They then use their so-called *structural learning* algorithm for semi-supervised learning of *one* classification task, by the artificial construction of 'related' problems on unlabeled data. This is done by creating problems of predicting *observable* features of the data and learning the structural regularization parameters from these 'auxiliary' problems and unlabeled data. More recently in (Ando and Zhang, 2007) they

[2] Whether or not a label is erroneous is independent of the feature values of the latter view.

showed that, with conditionally independent feature sets predicting from one set to the other allows the construction of a feature representation that leads to an effective semi-supervised learning algorithm. Our approach directly operates on the original feature space and can be viewed another justification for the algorithm in (Ando and Zhang, 2005).

Multiple Instance Learning (MIL) is a learning setting where training data is provided as positive and negative bags of samples (Dietterich et al., 1997). A negative bag contains only negative examples whereas a positive bag contains at least one positive example. Surrogate learning can be viewed as artificially constructing a MIL problem, with the *targets* acting as one positive bag and the *backgrounds* acting as one negative bag (Section 2.1). The class-conditional feature independence assumption for class $\mathbf{y} = 0$ translates to the identical and independent distribution of the negative samples in both bags.

4 Two Applications

We applied the surrogate learning algorithm to the problems of record linkage and paraphrase generation. As we shall see, the applications satisfy the assumptions in our second (100% recall) setting.

4.1 Record Linkage/ Entity Resolution

Record linkage is the process of identification and merging of records of the same entity in different databases or the unification of records in a single database, and constitutes an important component of data management. The reader is referred to (Winkler, 1995) for an overview of the record linkage problem, strategies and systems. In natural language processing record linkage problems arise during resolution of entities found in natural language text to a gazetteer.

Our problem consisted of merging each of ≈ 20000 physician records, which we call the *update database*, to the record of the same physician in a *master database* of $\approx 10^6$ records. The update database has fields that are absent in the master database and *vice versa*. The fields in common include the *name* (first, last and middle initial), several *address* fields, *phone*, *specialty*, and the *year-of-graduation*. Although the *last name* and *year-of-graduation* are consistent when present, the *address*, *specialty* and *phone* fields have several inconsistencies owing to different ways of writing the address, new addresses, different terms for the same specialty, missing fields, etc. However, the *name* and *year* alone are insufficient for disambiguation. We had access to ≈ 500 manually matched update records for training and evaluation (about 40 of these update records were labeled as unmatchable due to insufficient information).

The general approach to record linkage involves two steps: 1) *blocking*, where a small set of candidate records is retrieved from the master record database, which contains the correct match with high probability, and 2) *matching*, where the fields of the update records are compared to those of the candidates for scoring and selecting the match. We performed blocking by querying the master record database with the *last name* from the update record. Matching was done by scoring a feature vector of similarities over the various fields. The feature values were either binary (verifying the equality of a particular field in the update and a master record) or continuous (some kind of normalized string edit distance between fields like *street address*, *first name* etc.).

The surrogate learning solution to our matching problem was set up as follows. We designated the binary feature of equality of *year of graduation*[3] as the '100% recall' feature $\mathbf{x}_1$, and the remaining features are relegated to $\mathbf{x}_2$. The required conditions for surrogate learning are satisfied because 1) in our data it is highly unlikely for two records with different *year- of-graduation* to belong to the same physician and 2) if it is known that the update record and a master record belong to two *different* physicians, then knowing that they have the same (or different) *year-of-graduation* provides no information about the other features. Therefore all the feature vectors with the binary feature indicating equality of *year-of-graduation* are *targets* and the remaining are *backgrounds*.

First, we used feature vectors obtained from the records in all blocks from all 20000 update records to estimate the probability $P(\mathbf{x}_1|\mathbf{x}_2)$. We used lo-

[3] We believe that the equality of the middle intial would have worked just as well for $\mathbf{x}_1$.

Table 1: Precision and Recall for record linkage.

	Training proportion	Precision	Recall
Surrogate		0.96	0.95
Supervised	0.5	0.96	0.94
Supervised	0.2	0.96	0.91

gistic regression for this prediction task. For learning the logistic regression parameters, we discarded the feature vectors for which $\mathbf{x}_1$ was missing and performed mean imputation for the missing values of other features. Second, the probability $P(\mathbf{x}_1 = 1|\mathbf{y} = 0)$ (the probability that two different randomly chosen physicians have the same year of graduation) was estimated straightforwardly from the counts of the different years-of-graduation in the master record database.

These estimates were used to assign the score $P(\mathbf{y} = 1|\mathbf{x}_1 = 1, \mathbf{x}_2)$ to the records in a block (cf. Equation 4). The score of 0 is assigned to feature vectors which have $\mathbf{x}_1 = 0$. The only caveat is calculating the score for feature vectors that had missing $\mathbf{x}_1$. For such records we assign the score $P(\mathbf{y} = 1|\mathbf{x}_2) = P(\mathbf{y} = 1|\mathbf{x}_1 = 1, \mathbf{x}_2)P(\mathbf{x}_1 = 1|\mathbf{x}_2)$. We have estimates for both quantities on the right hand side. The highest scoring record in each block was flagged as a match if it exceeded some appropriate threshold.

We compared the results of the surrogate learning approach to a supervised logistic regression based matcher which used a portion of the manual matches for training and the remaining for testing. Table 1 shows the match precision and recall for both the surrogate learning and the supervised approaches. For the supervised algorithm, we show the results for the case where half the manually matched records were used for training and half for testing, as well as for the case where a fifth of the records of training and the remaining four-fifths for testing. In the latter case, every record participated in exactly one training fold but in four test folds.

The results indicate that the surrogate learner performs better matching by exploiting the unlabeled data than the supervised learner with insufficient training data. The results although not dramatic are still promising, considering that the surrogate learning approach used *none* of the manually matched records.

4.2 Paraphrase Generation for Event Extraction

Sentence classification is often a preprocessing step for event or relation extraction from text. One of the challenges posed by sentence classification is the diversity in the language for expressing the same event or relationship. We present a surrogate learning approach to generating paraphrases for expressing the *merger-acquisition* (MA) event between two organizations in financial news. Our goal is to find paraphrase sentences for the MA event from an unlabeled corpus of news articles, that might eventually be used to train a sentence classifier that discriminates between MA and non-MA sentences.

We assume that the unlabeled sentence corpus is time-stamped and named entity tagged with organizations. We further assume that a MA sentence must mention at least two organizations. Our approach to generate paraphrases is the following. We first extract all the so-called *source* sentences from the corpus that match a few high-precision seed patterns. An example of a seed pattern used for the MA event is '<ORG1> acquired <ORG2>' (where <ORG1> and <ORG2> are place holders for strings that have been tagged as organizations). An example of a *source* sentence that matches the seed is 'It was announced yesterday that <ORG>Google Inc.<ORG> acquired <ORG>Youtube <ORG>'. The purpose of the seed patterns is to produce pairs of participant organizations in an MA event with high precision.

We then extract every sentence in the corpus that contains at least two organizations, such that at least one of them matches an organization in the *source* sentences, and has a time-stamp within a two month time window of the matching *source* sentence. Of this set of sentences, all that contain *two* or more organizations from the *same source* sentence are designated as *target* sentences, and the rest are designated as *background* sentences.

We speculate that since an organization is unlikely to have a MA relationship with two different organizations in the same time period the *backgrounds* are unlikely to contain MA sentences, and moreover the language of the non-MA *target* sentences is

Table 2: Patterns used as seeds and the number of *source* sentences matching each seed.

	Seed pattern	# of *sources*
1	<ORG> acquired <ORG>	57
2	<ORG> bought <ORG>	70
3	offer for <ORG>	287
4	to buy <ORG>	396
5	merger with <ORG>	294

indistinguishable from that of the *background* sentences. To relate the approach to surrogate learning, we note that the binary "organization-pair equality" feature (both organizations in the current sentence being the same as those in a *source* sentence) serves as the '100% recall' feature $\mathbf{x}_1$. Word unigram, bigram and trigram features were used as $\mathbf{x}_2$. This setup satisfies the required conditions for surrogate learning because 1) if a sentence is about MA, the organization pair mentioned in it must be the same as that in a *source* sentence, (i.e., if *only* one of the organizations match those in a *source* sentence, the sentence is unlikely to be about MA) and 2) if an unlabeled sentence is non-MA, then knowing whether or not it shares an organization with a *source* does not provide any information about the language in the sentence.

If the original unlabeled corpus is sufficiently large, we expect the *target* set to cover most of the paraphrases for the MA event but may contain many non-MA sentences as well. The task of generating paraphrases involves filtering the *target* sentences that are non-MA and flagging the rest of the *targets* as paraphrases. This is done by constructing a classifier between the *targets* and *backgrounds*. The feature set used for this task was a bag of word unigrams, bigrams and trigrams, generated from the sentences and selected by ranking the n-grams by the divergence of their distributions in the *targets* and *backgrounds*. A support vector machine (SVM) was used to learn to classify between the *targets* and *backgrounds* and the sentences were ranked according to the score assigned by the SVM (which is a proxy for $P(\mathbf{x}_1 = 1|\mathbf{x}_2)$). We then thresholded the score to obtain the paraphrases.

Our approach is similar in principle to the 'Snowball' system proposed in (Agichtein and Gravano, 2000) for relation extraction. Similar to us, 'Snowball' looks for known participants in a relationship in an unlabeled corpus, and uses the newly discovered contexts to extract more participant tuples. However, unlike surrogate learning, which can use a rich set of features for ranking the *targets*, 'Snowball' scores the newly extracted contexts according to a single feature value which is confidence measure based only on the number of known participant tuples that are found in the context.

Example 2 below lists some sentences to illustrate the surrogate learning approach. Note that the *targets* may contain both MA and non-MA sentences but the *backgrounds* are unlikely to be MA.

Example 2

Seed Pattern

"offer for <ORG>"

Source Sentences

1. <ORG>US Airways<ORG> said Wednesday it will increase its **offer for <ORG>Delta<ORG>**.

Target Sentences (SVM score)

1.<ORG>US Airways<ORG> were to combine with a standalone <ORG>Delta<ORG>. (1.0008563)

2.<ORG>US Airways<ORG> argued that the nearly $10 billion acquisition of <ORG>Delta<ORG> would result in an efficiently run carrier that could offer low fares to fliers. (0.99958149)

3.<ORG>US Airways<ORG> is asking <ORG>Delta<ORG>'s official creditors committee to support postponing that hearing. (-0.99914371)

Background Sentences (SVM score)

1. The cities have made various overtures to <ORG>US Airways<ORG>, including a promise from <ORG>America West Airlines<ORG> and the former <ORG>US Airways<ORG>. (0.99957752)

2. <ORG>US Airways<ORG> shares rose 8 cents to close at $53.35 on the <ORG>New York Stock Exchange<ORG>. (-0.99906444)

We tested our algorithm on an unlabeled corpus of approximately 700000 financial news articles. We experimented with the five seed patterns shown in Table 2. We extracted a total of 870 *source* sentences from the five seeds. The number of *source* sentences matching each of the seeds is also shown in Table 2. Note that the numbers add to more than 870 because it is possible for a *source* sentence to match more than one seed.

The participants that were extracted from *sources*

Table 3: Precision/Recall of surrogate learning on the MA paraphrase problem for various thresholds. The baseline of using all the *targets* as paraphrases for MA has a precision of 66% and a recall of 100%.

Threshold	Precision	Recall
0.0	0.83	0.94
-0.2	0.82	0.95
-0.8	0.79	0.99

Table 4: Number of sentences found by surrogate learning matching each of the remaining seed patterns, when only one of the patterns was used as a seed. Each column is for one experiment with the corresponding pattern used as the seed. For example, when only the first pattern was used as the seed, we obtained 18 sentences that match the fourth pattern.

Seeds	1	2	3	4	5
1		2	2	5	1
2	5		6	7	5
3	4	6		152	103
4	18	16	93		57
5	3	9	195	57	

corresponded to approximately 12000 *target* sentences and approximately 120000 *background* sentences. For the purpose of evaluation, 500 randomly selected sentences from the *targets* were manually checked leading to 330 being tagged as MA and the remaining 170 as non-MA. This corresponds to a 66% precision of the *targets*.

We then ranked the *targets* according to the score assigned by the SVM trained to classify between the *targets* and *backgrounds*, and selected all the *targets* above a threshold as paraphrases for MA. Table 3 presents the precision and recall on the 500 manually tagged sentences as the threshold varies. The results indicate that our approach provides an effective way to rank the *target* sentences according to their likelihood of being about MA.

To evaluate the capability of the method to find paraphrases, we conducted five separate experiments using each pattern in Table 2 individually as a seed and counting the number of obtained sentences containing each of the other patterns (using a threshold of 0.0). These numbers are shown in the different columns of Table 4. Although new patterns are obtained, their distribution only roughly resembles the original distribution in the corpus. We attribute this to the correlation in the language used to describe a MA event based on its type (merger vs. acquisition, hostile takeover vs. seeking a buyer, etc.).

Finally we used the paraphrases, which were found by surrogate learning, to augment the training data for a MA sentence classifier and evaluated its accuracy. We first built a SVM classifier only on a portion of the labeled *targets* and classified the remaining. This approach yielded an accuracy of 76% on the test set (with two-fold cross validation). We then added all the *targets* scored above a threshold by surrogate learning as positive examples (4000 positive sentences in all were added), and all the *backgrounds* that scored below a low threshold as negative examples (27000 sentences), to the training data and repeated the two-fold cross validation. The classifier learned on the augmented training data improved the accuracy on the test data to 86% .

We believe that better designed features (than word n-grams) will provide paraphrases with higher precision and recall of the MA sentences found by surrogate learning. To apply our approach to a new event extraction problem, the design step also involves the selection of the $\mathbf{x}_1$ feature such that the *targets* and *backgrounds* satisfy our assumptions.

5 Conclusions

We presented surrogate learning – an easily implementable semi-supervised learning algorithm that can be applied when the features satisfy the required independence assumptions. We presented two applications, showed how the assumptions are satisfied, and presented empirical evidence for the efficacy of our algorithm. We have also applied surrogate learning to problems in information retrieval and document zoning. We expect that surrogate learning is sufficiently general to be applied in many NLP applications, if the features are carefully designed. We briefly note that a surrogate learning method based on regression and requiring only *mean independence* instead of full statistical independence can be derived using techniques similar to those in Section 2 – this modification is closely related to the problem and solution presented in (Quadrianto et al., 2008).

References

S. Abney. 2002. Bootstrapping. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367.

E. Agichtein and L. Gravano. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (ACM DL)*, pages 85–94, June, 2-7.

R. K. Ando and T. Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853.

R. K. Ando and T. Zhang. 2007. Two-view feature generation model for semi-supervised learning. In *ICML*, pages 25–32.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100.

O. Chapelle, B. Schölkopf, and A. Zien, editors. 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

R. O. Duda, P. E. Hart, and D. G. Stork. 2000. *Pattern Classification*. Wiley-Interscience Publication.

G. Nagy and G. L. Shelton. 1966. Self-corrective character recognition system. *IEEE Trans. Information Theory*, 12(2):215–222.

N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. 2008. Estimating labels from label proportions. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 776–783.

M. Seeger. 2000. Input-dependent regularization of conditional density models. Technical report, Institute for ANC, Edinburgh, UK. See `http://www.dai.ed.ac.uk/~seeger/papers.html`.

W. E. Winkler. 1995. Matching and record linkage. In *Business Survey Methods*, pages 355–384. Wiley.