2005

Cost-Sensitive Learning for Confidential Access Control

Young Woo Seo
Carnegie Mellon University

Drew Bagnell
Carnegie Mellon University

Katia Sycara
Carnegie Mellon University

Cost-Sensitive Learning for Confidential Access Control

Young-Woo Seo Drew Bagnell Katia Sycara

CMU-RI-TR-05-31

June 2005

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

It is common to control access to critical information based on the need-to-know principle; The requests for access are authorized only if the content of the requested information is relevant to the requester's project. We formulate such a dichotomous decision in a machine learning framework. Although the cost for misclassifying examples should be differentiated according to their importance, the best-performing error-minimizing classifiers do not have ways of incorporating the cost information into their learning processes. In order to handle the cost effectively, we apply two cost-sensitive learning methods to the problem of the confidential access control and compare their usefulness with those of error-minimizing classifiers. We devise a new metric for assigning cost to any datasets. From the comparison of the cost-sensitive classifiers with error-minimizing classifiers, we find that costing demonstrates the best performance in that it minimizes the cost for misclassifying the examples and the false positive using a relatively small amount of training data.

Contents

1 Introduction **1**

2 Cost-Sensitive Classification **2**
2.1 An Illustrative Example 2
2.2 Wrappers for Cost-Sensitive Classification 4
2.2.1 Costing 5
2.2.2 Meta-costing 6

3 Binary Classification For Supporting Critical Decision **6**
3.1 Linear Discriminant Analysis 6
3.2 Logistic Regression 7
3.3 Support Vector Machines 8
3.4 Naive Bayes Classifier 8

4 Experiments **9**
4.1 Cost Assignment 10
4.2 Experimental Results 12

5 Related Work **14**

6 Conclusion and Future Work **15**

7 Acknowledgement **16**

1 Introduction

Securing information from unauthorized accesses is very important in an information-rich society. For example, project managers want to protect their trade secrets from employees in other departments as well as outsiders. For the purpose of indexing and security, confidential information is grouped into containers based on the similarity of their contents or similar levels of confidentiality. A secure repository (e.g., a secured database) holds all these containers encompassed by a limited access system. Requests to access the confidential information may occur, for example, when an employee is assigned to a new project and needs to access background knowledge. A set of access control lists (ACL) might be compiled manually to control those requests. Each item of confidential information is associated with an ACL, which ensures a corresponding level of security and can be accessed by anyone who has been authorized. However this approach has a crucial security weakness in that a user who is authorized to a segment of confidential information in a container is actually able to access the entire container. For example, an employee, who is authorized only to look at a progress report on the development of new technology, is able to access the information about a financial plan for that project; the two pieces of information are about the same project and hence are held in the same container. Therefore the supervisor of the collection of confidential information will either hand select only those documents that he will let the user see, or completely bar access to the entire collection rather than risk exposing documents that should not be exposed.

Furthermore, this approach is inflexible. It does not allow easy adjustment to frequent changes of a user's task assignment. Project assignments for an employee may be changed quite often and hence the employee needs to access confidential information related to the newly assigned project. In addition, access to a previously assigned project may need to be revoked. In order to ensure that authorized access is granted and unauthorized access is denied, the ACLs for all information associated with the project must be updated according to the rights and the permissions of employees assigned to the project.

As a solution for these problems, we developed a multi-agent system that handles the authorization of requests for confidential information as a binary classification problem [13]. Instead of relying on coarse-grained ACLs and handpicked information, our system compares the content of requested confidential information with the content of the requester's project and authorizes the request only if the two are relevant. In the case of either acceptance or rejection, the event can be logged for security audits and alarms. By doing this, our system allows the supervisor a means of specifying subsets of per-user and per-task access control policies and a way to automatically enforce them. Since the proposed system learns the supervisor's decision criteria based on a small number of supervisor-provided examples, the supervisor need not identify all relevant information. Through our proposed system, it then becomes possible for the supervisor to define, assign, and enforce a security policy for a particular subset of confidential information.

Although our approach showed a relatively good performance [13], we believe there is a room for improvement. Previously we made use of five different error-minimizing classifiers for authorizing the requests to access confidential information.

We believe that we can improve our results by taking into consideration the cost caused by misclassification. In particular, it is undesirable to use an error-minimizing classification method, which treats all mis-classification costs equally, for this scenario because primarily it classifies every example as belonging to the most probable class. For example, suppose that there are 100 medical records that are actually comprised of 5 cancer records and 95 cold records. Without considering the cost for misclassification (i.e., diagnosis), an error-minimizing classifier could simply achieve the lower error rate by ignoring the minority class, even though the actual result of misdiagnosis on cancer is far worse than that of cold (e.g., the cost will be really high).

In this paper we would like to test the effectiveness of cost-sensitive learning for the problem of confidential access control. Section 2 compares cost-sensitive classification with error-minimizing classification in terms of the optimal decision boundary and details two approaches for cost-sensitive learning. Section 3 describes three different classification methods as candidates for the process of confidential access control. Section 4 describes experimental settings and empirical evaluation of cost-sensitive learners. Section 5 presents related work and section 6 presents conclusions and future work, respectively.

2 Cost-Sensitive Classification

In the previous section, we mentioned briefly the reason why a cost-sensitive classifier is better suited for the problem of confidential access control than an error-minimizing classifier. This section formalizes the principle in terms of the optimal decision boundary for a binary classification task with univariate data.

2.1 An Illustrative Example

A classification method is a decision rule to assign one of (or more than one) predefined classes to given examples. Some of them produce a continuous output whereas others produce a discrete class label. The optimal decision boundary is a decision criteria that allows a classifier to produce the best performance.

Let us consider a hypothetical example in figure 1 which shows two classes with overlapping boundaries due to their intrinsic randomness – their actual values are random variables. In this example, the probability density for each class is normal, that is, $p(class = 0|x) \sim N(\mu_0, \sigma_0^2)$ and $p(class = 1|x) \sim N(\mu_1, \sigma_1^2)$[1].

If the cost for misclassification is equal, where is the optimal decision boundary (xe^*) for a binary classification? Assuming that we know how two probability densities are distributed, it is relatively easy to compute the optimal decision boundary. Formally, let P be the probability of class 1 given an example x.

$$\begin{aligned} P(x|class = 1) &= P(x|class = 0) \\ P &= 1 - P \\ P &= 0.5 \end{aligned}$$

[1] $\mu_0 = 0.3500, \sigma_0 = 0.1448, \mu_1 = 0.7000, \sigma_1 = 0.1736$

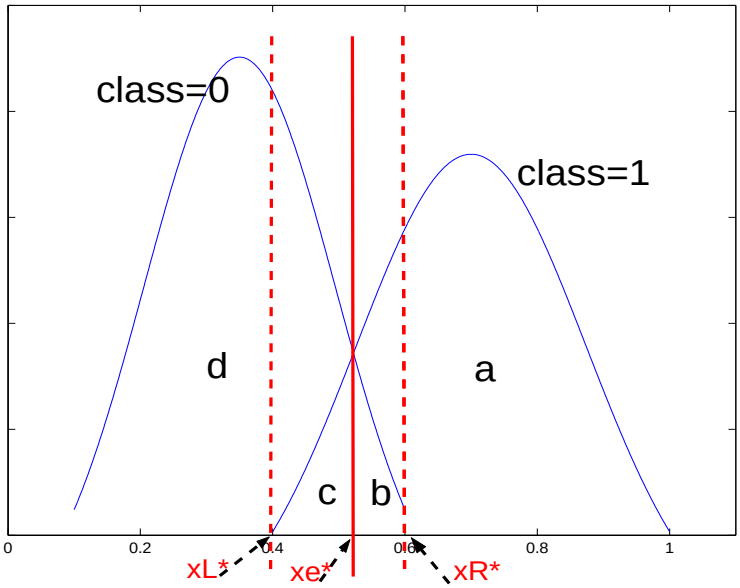

Figure 1: The optimal decision boundary for a binary classification will be determined by considering the misclassification cost.

The probability of an example x belonging to class 1 is 0.5, meaning that the optimal decision boundary lies in the center of two class distributions (i.e., $xe^* = 0.52$). Therefore an example randomly generated will be assigned to class 1 if its value is greater than 0.52. The solid line represents this optimal decision boundary, assuming the cost for misclassifying is equal.

Pivoting the optimal boundary, a classifier could have four possible classification outcomes; "a" is "true positive" that an example, x, belongs to "class 1" and it is classified as "class 1." "c" is "false negative" if x is classified as "class 0." "d" is "true negative" if x belongs to "class 0" and is classified as "class 0." Finally, "b" is "false positive" if x is classified as positive [10]. Table 1 captures this information as well as the cost (λ_{ij}) involved in those four outcomes. Particularly, λ_{ij} is the cost for classifying an example belonging to j as i.

It is reasonable to evaluate the performance of a classifier by computing the area under the regions' boundaries. In particular, the true area for class 1 is the sum of "a" and "c." If a classifier produces results like those in table 1 (e.g., "a" and "b" for "class 1" and "c" and "d" for "class 0"), then the false negative of this classifier is roughly 15% (i.e., percentage of "c" out of the whole area of "class 1") and the false positive is 10% (i.e., 10% of class 0), respectively.

Where then would be the optimal decision boundary if the cost for misclassifying is unequal. Let us assume that text documents belonging to "class 0" and "class 1" are confidential information of which careless release may have a damaging effect. An employee is newly assigned to a project for which records are stored in a secured

	true $class = 1$	true $class = 0$
output $class = 1$	a (λ_{11})	b (λ_{10})
output $class = 0$	c (λ_{01})	d (λ_{00})

Table 1: Four possible outcomes and their costs for a binary classification are presented.

repository and are labeled as "class 1." Since the documents belonging to "class 0" are about different projects, the employee is supposed to access to only documents in "class 1" for understanding background knowledge of the project.

Assuming that no cost is assigned to the correct classification ($\lambda_{11} = \lambda_{00} = 0$), the costs for two errors should be considered carefully for providing a reliable confidential access control; false negative (λ_{01}) – reject the valid request (e.g., reject the request that the employee asks to access a "class 1" document by predicting the requested document as "class 0"); false positive (λ_{10}) – accept the invalid request (e.g., accept the request that the employee asks to access a "class 0" document by predicting the requested document as "class 1"). As the results from such invalid authorizations, a false negative causes the employee to be inconvenienced because he is not able to access need-to-know information. However, not approving valid requests does not cause a serious problem from the security perspective. On the contrary, a false positive is a serious problem because confidential information, which should not be revealed, can be accessed. Therefore, for a need-to-know basis confidential authorization, the cost for false positive is much higher than that of false negative. Thus it is reasonable to re-locate the decision boundary for uniform-cost (i.e., solid line in the figure 1), in order to minimize the cost for misclassifications. For example, if the cost of false positive is higher than that of false negative, the decision line should be moved toward to the right (i.e., xR^*). Two dashed lines in the figure 1 represent the optimal decision boundaries for non-uniform misclassification cost assigned to each example.

However a tradeoff must be considered because choosing one of the extremes (e.g., xL^* or xR^*) will sacrifice the error that is not considered. In particular, the classifier could reduce the false negative close to zero if we would choose xL^* as a decision line, but with higher false positive. If either of extremes is not the solution, the optimal decision line should be chosen somewhere between extremes by considering the tradeoff:

$$\begin{aligned} cost_1 P(y=1|\vec{x}) &= cost_2 P(y=0|\vec{x}) \\ cost_1 P &= cost_2(1-P) \\ cost_1 P &= cost_2 - cost_2 P \\ P &= \frac{cost_2}{cost_1 + cost_2} \end{aligned}$$

2.2 Wrappers for Cost-Sensitive Classification

In the problem of unequal misclassification cost, the example space is optimally divided into $|C|$ regions so that class j is the optimal (i.e., least-cost) prediction in region j. The goal of cost-sensitive learning is to find the boundary between these regions. Obviously the misclassification cost, particularly a loss matrix (e.g. table 1), is the dominant factor for the optimal boundaries. That is, the region where j must be predicted will expand at the expense of the regions of other classes if misclassifying examples of class j is more expensive relative to misclassifying others, even though the class probabilities remain unchanged.

There have been two major approaches for cost-sensitive learning. The first one is a glass-box approach that modifies particular error-minimizing classifiers cost-sensitive

[1], [16]. The second one is a black-box approach that converts arbitrary error-minimizing classifiers into cost-sensitive ones [19], [2].

In this paper, we utilize two methods in the black-box approach for cost-sensitive learning: costing [19] and metacost [2]. A black-box approach for cost-sensitive learning makes any error-minimizing learning method carry out cost-sensitive learning. In particular, they make use of sampling techniques that change the original example distribution D to $\hat{D}$ by incorporating into it the relative cost of each instance. Then they make any cost-insensitive error-minimizing classifiers perform expected cost minimization on the newly generated distribution, $\hat{D}$. According to a given cost matrix, this changes the proportion of a certain class by re-sampling of the original examples instead of modifying the learner's rule.

2.2.1 Costing

Costing (cost proportionate rejection sampling with aggregation) is a wrapper for cost-sensitive learning that trains a set of error-minimizing classifiers by a distribution, which is the original distribution with the relative cost of each example, and outputs a final classifier by taking the average over all learned classifiers [19]. It assumes that changing the original example distribution D to another $\hat{D}$, by combining it with the cost information, makes any error minimizing classifier accomplish expected cost minimization on the original distribution. Costing is comprised of two processes: rejection sampling and bagging. Rejection sampling has been used to generate independently and identically distributed (i.i.d.) samples that are used as a proxy distribution to achieve simulation from the target distribution. To this end, it requires a density function $g(x)$ and a constant $M > 1$, satisfying the "envelope property"

$$\pi(x) \leq Mg(x)$$

Given the a density function $g(x)$ satisfying the "envelope property", rejection sampling works as follows: draw x from $g(x)$ and a sample u from a uniform distribution $U, \forall u \in [0, 1]$ and accept x if $u < \frac{\pi(x)}{Mg(x)}$. Otherwise reject the value of x and repeat the sampling step [11]. The accepted values are regarded as a realization of $\pi(x)$. In particular, suppose we run the sampling N times, and can estimate μ by using the N accepted samples because those samples are i.i.d. samples from $\pi(x)$.

Rejection sampling for costing assigns each example in the original distribution with a relative cost [2] and draws a random number $r \in [0, 1]$ from a uniform distribution U. It will keep the example if $r > \frac{c}{Z}$. Otherwise it discards the example and continues sampling until a certain criteria is satisfied. The accepted examples are regarded as a realization of the altered distribution, $\hat{D}, \hat{D} = \{S_1^{'}, S_2^{'}, ..., S_k^{'}\}$. With the altered distribution, $\hat{D}$, costing trains k different hypotheses, $h_i \equiv Learn(S_i^{'})$, and predicts the label of an test example, $\vec{x}$, by combining those hypotheses, $h(\vec{x}) = sign\left(\sum_{i=1}^{k} h_i(\vec{x})\right)$.

[2] $\hat{x}_i = \frac{c}{Z} \times x_i$, where c is a cost assigned to x_i and Z is $\max_{c \in S} c$.

2.2.2 Meta-costing

The MetaCost is another method for converting an error-minimizing classifier into cost-sensitive classifier by re-sampling [2]. The underlying assumption is that an ordinary classifier for error-minimization could learn the optimal decision boundary based on the cost matrix if each training example is relabeled with the cost. The learning process of MetaCost is comprised of two processes: bagging and retraining the classifiers with cost. In particular, it generates a set of samples with replacement from the training set and estimates the class of each instance by taking the average of votes over all the trained classifiers. Then the MetaCost re-labels each training example with the estimated optimal class and re-trains the classifier to the relabeled training set.

$$R(i|\vec{x}) = \arg\min_j \{P(j|\vec{x})C(i,j)\}$$

where $R(i|\vec{x})$ is the expected cost of predicting that $\vec{x}$ belongs to the ith class and $P(j|\vec{x})$ is the Bayes optimal classification.

3 Binary Classification For Supporting Critical Decision

Our goal is to develop a reliable process for confidential access control based on the need-to-know principle; the request for access to a unit of confidential information is accepted only if the content of the requested item is relevant to the requester's task. It is reasonable to verify whether or not the content of a requested confidential item is associated with the content of a requester's project because the requester only needs to know information related to his/her project in order to conduct the given task. In other words, a request for confidential that the requester does not "need to know" is undoubtedly rejected. To this end, we model such a dichotomous decision (i.e., to reject or accept the request) in a machine learning framework. We choose four different classification methods, linear discriminant analysis, logistic regression, support vector machines, and naive Bayes classifier, because of their relative good performance, particularly in text classification [5], [12], [14], [9].

3.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a well known method in statistical pattern recognition that projects the observed patterns into a low dimensional space in which the classes are well separated [6]. In particular, LDA produces an optimal linear discriminant function $\mathbf{f}(\mathbf{x}) = W^T\mathbf{x}$ which maps the input example into the classification space in which the class identification of this sample is decided based on some metric such as Euclidean distance and Mahalanobis distance. A typical LDA implementation is carried out via scatter matrix analysis. We compute the within and between-class scatter matrices as follows:

$$S_w = \frac{1}{M}\sum_{i=1}^{M} P(C_i)\mathbf{\Sigma}_i \quad (1)$$

$$S_b = \frac{1}{M}\sum_{i=1}^{M}(\mu_i - \mu)(\mu_i - \mu)^T \quad (2)$$

where μ_i and $\mathbf{\Sigma}_i$ is the mean vector and covariance matrix of the ith class, S_w is the within-class scatter matrix showing the average scatter $\mathbf{\Sigma}_i$ of the sample vectors $\vec{x}$ of different class C_i around their respective mean μ_i, S_b is the between-class scatter matrix, representing the scatter of the conditional mean vectors μ_i's around the overall mean vector μ. Various measures are available for quantifying the discriminatory power, the commonly used one being [6]:

$$J(W) = \frac{||W^T S_w W||}{||W^T S_b W||}$$

where W is the optimal discrimination projection and can be obtained via solving the generalized eigenvalue problem:

$$S_b W = \lambda S_w W$$

The distance measure used in the matching could be a simple Euclidean or Mahalanobis. However for our case – a binary classification whether a document belongs to the need-to-know confidential or not – Euclidean distance is used because the maximum rank of S_b is $|C| - 1$, where $|C|$ is the number of classes, meaning that LDA cannot produce more than $|C| - 1$ features. LDA has been used in [12] as a text classification method and in [14] as a feature selection method.

3.2 Logistic Regression

Logistic regression is a statistical technique for modeling a binary response variable by a linear combination of one or more features, using a logit link function

$$\begin{aligned} P(class = 1|\mathbf{w}, \mathbf{x}) &= \psi\left(\mathbf{w}^T\mathbf{x}\right), \\ &\text{where,} \\ \psi(\mathbf{w}^T\mathbf{x}) &= \frac{\exp(\mathbf{w}^T\mathbf{x})}{1+\exp(\mathbf{w}^T\mathbf{x})} = \frac{1}{1+\exp^{-\mathbf{w}^T\mathbf{x}}} \end{aligned}$$

The Bayesian approach to the logistic regression assumes gaussian priors, $p(class = 0|x) \sim N(\mu_0, \sigma_0^2$ (and $p(class = 1|x) \sim N(\mu_1, \sigma_1^2)$. In order to find the Maximum A posterior Probability (MAP) estimate of $\mathbf{w}$:

$$\begin{aligned} L(\mathbf{w}) &= \arg\max_j \{P(\mathbf{w}_j|D)\} \\ &= \arg\max_j \left\{ P(\mathbf{w}_j) \prod_{i=1}^{n} \frac{1}{1+\exp(-\mathbf{w}_j^T\mathbf{x}_i y_i)} \right\} \\ &= \arg\max_j \left\{ \ln P(\mathbf{w}_j) - \sum_{i}^{n} \ln(1+\exp(-\mathbf{w}_j^T\mathbf{x}_i y_i)) \right\} \end{aligned}$$

where $P(\vec{w})$ is the prior on $\vec{w}$ and n is the number of the training examples.

3.3 Support Vector Machines

Support Vector Machines (SVMs) learns the parameters $\mathbf{w}$ and b specifying a linear decision rule $h(x) = sign(\mathbf{w} \cdot x + b)$, so that the smallest distance between each training example and the decision boundary (i.e., margin) is maximized [15]. It works by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \tfrac{1}{2}\mathbf{w}^T \cdot \mathbf{w} + C\sum_{i=1}^{n} \xi_i \\ &\text{subject to} \quad \forall i, y_i \left(\mathbf{w}^T \phi(x_i) + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

Here training vector, x_i, is mapped into a higher dimensional space by the function ϕ. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. The constraints require that all examples in the training set are classified correctly up to some slack variable ξ_i. If a training example lies on the wrong side of the decision boundary, the corresponding ξ_i is greater than 1. Therefore, $\sum_{i=1}^{n} \xi_i$ is an upper bound on the number of training errors. The factor C is a parameter that allows one to trade off training error and model complexity.

3.4 Naive Bayes Classifier

A Bayesian learning framework assumes that the examples were generated by a parametric model and uses training data to compute Bayes-optimal estimates of the model parameters. With these estimates, it classifies new test examples using Bayes' rule to calculate the posterior probability that a class would have generated the test example in question. Then classification is carried out by selecting the most probable class. In addition to this framework, the naive Bayes classification assumes that all attributes of the examples are independent of each other given the context of the class.

The naive Bayes classifier we used is a multinomial model that represents an example as the set of attribute occurrences from the training set. It is assumed that the individual attribute occurrences to be the "events" and the example to be the collection of attribute events. It is known to perform better than a multi-variate Bernouli model, where an example is regarded as "event" that is consisted of the absence or presence of attributes, because of capturing the frequency of the attributes [9].

Naive Bayes classifier predicts the class c_j that maximizes the posterior probability, $P(c_j|\mathbf{x})$, for an example vector $\mathbf{x}$, under the attribute independence assumption.

$$\begin{aligned} P(c|\mathbf{x}_i) &= \arg\max_j \{P(c_j)P(\mathbf{x}_i|c_j)/P(\mathbf{x})\} \\ &= \arg\max_j \{P(c_j)P(\mathbf{x}_i|c_j)\} \\ &= \arg\max_j \left\{ P(c_j) \prod_{k=1}^{|A|} P(a_k|c_j)^{n_{i,k}} \right\} \\ &= \arg\max_j \left\{ \log P(c_j) + \sum_{k=1}^{|A|} \log P(a_k|c_j)^{n_{i,k}} \right\} \end{aligned}$$

$$= \quad \arg\max_j \left\{ \log \hat{P}(c_j) + \sum_{k=1}^{|A|} n_{i,k} \log \hat{P}(a_k|c_j) \right\}$$

where $|A|$ is the total number of attributes, $n_{i,k}$ is the frequency of the kth attribute in the ith example, $\hat{P}(c_j) = \frac{X(c_j)}{X}$, and $\hat{P}(a_k|c_j) = \frac{1+n_{j,k}}{|A|+\sum_{k\in|A|} n_{j,k}}$.

Although the assumption on the feature independence is unrealistic in a real-world problem, the naive Bayesian classification has been shown to be surprisingly effective and is computationally efficient [9]. In other words, training such a classifier only requires time that is linear in the number of features and data instances, meaning that they do not use word combinations as predictors and are thus far more efficient than the exponential non-Bayes approaches.

4 Experiments

As we described earlier, the scenario which we are particularly interested in is a process of confidential access control based on the need-to-know principle. The purpose of the experiments is two-fold; to find a good classification method that minimizes the cost and the false positive rate while holding the false negative rate reasonably low; to verify that the wrappers for cost-sensitive learning reduce the total cost loss in comparison with error-minimizing classifiers. From these objectives, three performance metrics are primarily used to measure the usefulness of classifiers; false negative, defined as $fn = \frac{c}{a+c}$ by using the values in the table 1, false positive, $fp = \frac{b}{b+d}$, and cost for misclassification. These metrics are better matched to our purpose than conventional measures based on precision-recall because we are interested in primarily reducing the error and the cost. Moreover, two error measures are not sensitive to changes of class frequency whereas the precision and recall are sensitive to the frequencies of the target classes [4].

Since there are no datasets available that are comprised of confidential information, we choose the Reuters-21578 document collections for experiments. This data set, which consists of world news stories from 1987, has become a benchmark in text categorization evaluations. It has been partially labelled by experts with respect to a list of categories. These categories have been grouped into super-categories of people, topics, places, organisations etc. The category distribution is skewed: the most common category has a training-set frequency of 2,877, but 82% of the categories have less than 100 instances and 33% of the categories have less than 10 instances. There are 135 overlapping topic categories. Since it is a binary classification task where each document has an exclusive category (i.e., either positive or negative), we discarded documents that are assigned no topic or multiple topics. Moreover, classes with frequencies less than 10 are discarded. The resulting data set is comprised of 9,854 documents as a training set and 4,274 documents as a test set with 67 classes (topics).

Each document is represented by a multi-dimensional vector of which size is determined by the size of the vocabulary and its element corresponds to a word in the vocabulary. The vocabulary was constructed by discarding stop words, too frequent

words, and rarely occurring words [3] without stemming. After these processes, the size of the original vocabulary (i.e., a set of unique unigrams) was reduced to 7,114 from 23,918. In order to reduce the dimensionality further, we tested three different feature selection methods such as χ^2 statistics, information gain, and point-wise mutual information. We found the performance of χ^2 method best. This replicates results reported in other research works [18], [17]. The dimension of the feature space is finally set to 1,000. Previous work on the Reuters-21578 dataset showed that such a drastic reduction of the feature space's dimension does not degrade the performance [5], [9]. Each document is then represented by using those selected 1,000 words and their weights are computed by:

$$w_{i,k} = \left(\frac{tf_{i,k}}{tf_{i,k} + 0.5 + 1.5 \times \frac{dl_i}{\text{ave dl}}} \right) \times \left(\log_2 \frac{N}{df_k} + 1.0 \right)$$

where $tf_{i,k}$ is the frequency of the kth word in the ith document, dl_i is the ith document length, $ave\ dl$ is the average document length in the training documents, N is the total number of training documents, df_k is the document frequency of the kth word. The final size of the word-by-document matrix is 1000×9854, which is reasonably smaller than the original matrix, 7114×9854. For succinct representation of this matrix, we tested two different techniques for dimension reduction: principal component analysis (PCA) and LDA/PCA. Since there is a noticeable performance difference between two techniques in representing 90% of the total variance [4]in the covariance matrix, we used PCA alone for concise representation of the word-by-document matrix.

The experimental setting is as follows. All the documents are regarded as confidential and accordingly they are kept in a secured container, ensuring that authorized users are only allowed to access. Documents belonging to the selected category are regarded as confidential information that the requester needs to know. Conversely the rest of test documents are confidential information that should not be revealed. A false positive occurs when the system accepts a request that should have not been accepted whereas a false negative occurs when the system rejects a request that should have been accepted. From the security perspective, it is more tolerable to have an authorization process with a high false negative rate than one with a high false positive rate.

4.1 Cost Assignment

According to the class assignment – not the original category label, but the artificially assigned class label, such as need-to-know confidential or otherwise (simply, positive or negative) – each of the documents in both the training and testing sets is assigned by a cost, ensuring that the mis-classification cost of a need-to-know confidential information is higher than that of remaining confidential (i.e., $\lambda_{10} > \lambda_{01},\ \lambda_{10} > \lambda_{00},\ \lambda_{01} > \lambda_{11}$) [3]. Otherwise unreasonable assignment of cost leads a classifier to always predict

[3]This is done by removing words if their document frequency is less than a threshold of "rarely occurred" (e.g., 3) or is greater than the threshold (e.g., 500) of "too frequent."

[4]$\hat{D} = \vec{e}_k^T D$, where, $\hat{D} = k \times n, D = m \times n, k \ll m$, $k = \frac{\lambda_i}{\sum_j \lambda_j} \geq 90\%$ of variance in $\Sigma, \Sigma\lambda = \vec{e}\lambda$.

the dominant class, regardless of what the true class is. In particular, let us assume that the cost of first row in the table 1 is greater than that of the second one (i.e., $\lambda_{1j} \geq \lambda_{0j}$). A classifier then always predicts "class=0" regardless of its learned rules (e.g., posterior probability distribution).

Since the Reuters-21578 document collection does not have cost information, we devised a heuristic for cost assignment. It complies with our idea; firstly, there is a cost involved in incorrect classification; secondly, the higher cost is assigned to a false positive than a false negative. Particularly, the cost for misclassifying a document, $\mathbf{d}_i$, is computed by:

$$cost(\mathbf{d}_i) = \begin{cases} [s, s + |c_j|] \text{ if } \mathbf{d}_i \in c_j \text{ and } c_j = \text{positive} \\ \left[0, \frac{\sum_{s \in positive} cost(\mathbf{d}_s)}{|\text{number of negative documents}|}\right] \text{ Otherwise} \end{cases}$$

where $s = \ln\left(\frac{N}{|c_j|}\right) \times 100$, N is the total number of documents and $|c_j|$ is the number of documents belonging to the jth category. The total cost for misclassification is added to the cost of confidential documents misclassed if a classifier is not able to predict any of the positive cases, in order to prevent the case that a low cost is simply achieved by ignoring the class with a low frequency. For example, there are 10 out of 10,000 documents belonging to the positive class. The cost assignment ensures that the total cost for misclassifying those 10 examples should be either equal to or higher than that of the remaining documents [5]. This heuristic method is intended to prevent the case that a low cost can be achieved simply by ignoring the minority class. In particular, there are 10 examples out of 10,000 examples belonging to the positive class. The cost assignment ensures that the total cost for misclassifying those 10 examples should be either equals to or higher than that of remaining documents [6] When this is the case, a low error rate can be achieved simply by ignoring the confidential class. In particular, a dumb classifier will achieve 99% accuracy by simply predicting all documents as negative and it will pay a half of the total cost for its incorrect classification. Obviously this should be avoided. To this end, the total cost for misclassification is added by the cost of confidential documents misclassified if a classifier is not able to predict any positive cases. The previous dumb classifier will be paying 13815.6 because it classifies all positive examples incorrectly, even though it does all negative examples correctly. The total cost is computed by summing the misclassification cost of positive and negative examples. The cost for misclassifying positive case will impose to a classifier if it is only able to classify all non-confidential correctly. It is reasonable that a classifier, which predicts the label of all positive documents incorrectly and does all negative correctly, will be eventually paying the same amount of cost paid by another classifier that classifies all documents incorrectly.

This assignment method works for both cost assigned for each example and cost assigned for each case (e.g., false alarm and miss). For the per example cost-sensitive

[5] For this case, the cost for misclassifying a positive document is 690.67 ($\ln\left(\frac{9990}{10}\right) \times 100 = 690.6755$) and the sum of the cost is 6906.755 (690.6755×10). Accordingly the cost of misclassification of a negative document is 0.6913 ($\frac{6906.755}{9990} = 0.6913$) and the cost sums to 6906.087.

[6] For this case, the cost for misclassifying a positive document is 690.78 and the sum of the cost is 6907.8 (690×10). Accordingly the cost of misclassification of a negative document is 0.6914 and the cost sums to 6907.8.

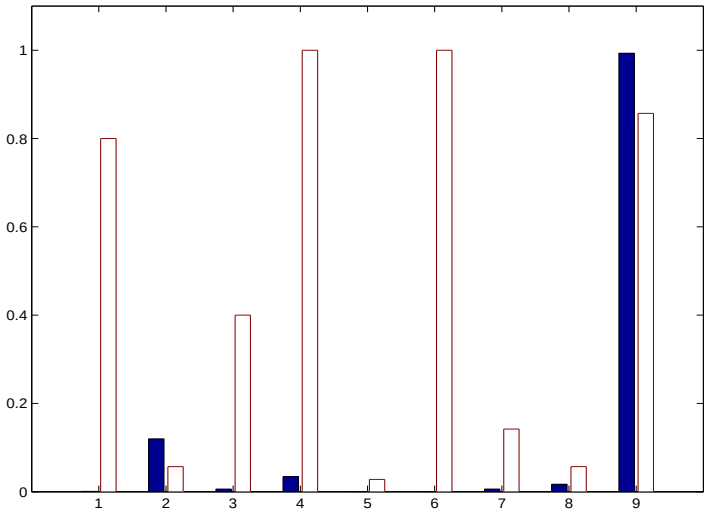

Figure 2: A pair of of false positive (filled bar) and false negative (empty bar) for "livestock" category is presented.

learning, the misclassification cost should be paid when a positive is classified as a negative. For the previous example, the cost for false alarm (i.e., λ_{10} in the table 1) is 690.67 whereas the cost for miss (i.e., λ_{01}) is 0.6913, respectively.

4.2 Experimental Results

We choose the five different categories as representative categories according to their category frequencies: small (livestock and corn), medium (interest), and large (acq and earn). There are 70 % of documents in a category used as "training" and the remaining 30 % documents are used for "testing", respectively. There are nine different classifiers tested: LDA, LR, and SVMs, and the combination of those three classifiers with two wrappers for cost-sensitive learning: metacost (MC) and costing [7]. A binary classifier was trained for each of the selected categories by considering the category as positive with the rest of the data as negative examples. We made use of the LIBSVM[8] and tested three different kernels, such as linear, polynomial, and Gaussian. The Gaussian kernel ($width = \frac{1}{\text{max feature dimension}}$) was chosen due to its best performance and the different cost factors are assigned [9], $C = 10 \sim 100$. Those values are chosen optimally by 10-fold cross validation.

The experimental results are primarily analyzed by "false positive," "false negative," and "cost." The procedure of experiments is as follows: firstly, pick one of five selected categories; secondly, assign the cost to each of examples according to its importance using the heuristic described in section 4.1; then, train each of nine classifiers by training examples with cost; finally compute three performance measure by using the contingency table.

Figure thru 2 to 6 show pairs of false positive and false negative for each of the

[7] The results of naive Bayes classifiers were removed due to its poor performance.

[8] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[9] The cost of constrain violation is set to 100 if there are relatively small amount of positive examples available. Otherwise it is set to about 10.

Methods	livestock (114)	corn (253)	interest (513)	acq (2448)	earn (3987)
SVM	13967	66453	54065	83141	108108
SVM (w/ costing)	4035 ± 30	8851 ± 52	9058 ± 159	40009 ± 252	96007 ± 331
SVM (w/ mc)	7147 ± 50	23596 ± 64	32011 ± 321	194165 ± 451	228612 ± 453
LR	35809	32759	60031	349080	710631
LR (w/ costing)	$\mathbf{484 \pm 11}$	$\mathbf{1333 \pm 44}$	29614 ± 110	$\mathbf{606 \pm 145}$	$\mathbf{2521 \pm 191}$
LR (w/ mc)	34980 ± 35	32759 ± 79	60374 ± 154	386859 ± 1185	788819 ± 263
LDA	2638	66453	124733	591300	908690
LDA (w/ costing)	1461 ± 28	6092 ± 89	$\mathbf{7301 \pm 152}$	39354 ± 205	41478 ± 159
LDA (w/ mc)	40079 ± 57	45778 ± 71	8955 ± 157	51789 ± 285	54084 ± 244
Cost for base line	42625	79084	139113	591357	1090498

Table 2: The cost by nine different classifiers are presented. The values in bold face are the best for corresponding category.

selected categories by nine different classifiers, which are numbered from the left to right: SVM (1), SVM with costing (2), SVM with metacost (3), LR (4), LR with costing (5), LR with metacost (6), LDA (7), LDA with costing (8), and LDA with metacost (9), respectively.

Except the "interest" category, LR with costing showed the best results that minimize false positive while holding false negative low. In particular, for the "livestock" category, LR trained by only 18% training data (i.e., 1781 out of 9854 documents) resulted 0% false positive and 2.8% false negative rate. For the costing, we carried out five different sampling trials for each category (i.e., 1, 3, 5, 10, and 15) and represented the trial for the best performance. For this category, a newly generated distribution by 10 rejection sampling trials is used to achieve this result. Each resampled set has only about 178 documents. LDA with costing showed the smallest error for the "interest" category that is comprised of 5.6% false positive and 4.5% false negative.

Table 2 replicates this trend in terms of the total cost for misclassification. The number in parenthesis next to topic name is the total number of text documents belonging to that category. The results reported for costing and metacost are the average of 5 different runs. The bottom line entitled "cost for base line" is the cost for a category if a classifier classifies all the testing examples incorrectly (e.g., a classifier for "livestock" category will cause 42625 for misclassification cost if it classifies all incorrectly). For the "earn" category, LR with costing caused only 0.002 out of the total cost (2521 out of 1090498). For the remaining categories, the best-performer paid only less than 0.05 out of the total cost.

From the comparison with error-minimizing classifiers, the costing proved its effectiveness in that it requires relatively small amount of training data for a better performance. For the "corn" category, LR with costing, which only used 10% of the training data (i.e., 986 out of 9854 documents) showed the best result in terms of the smallest loss (1333 out of 79084), zero false positive, and lower false negative rate (0.039). The LR classifier was trained by a sample set by three rejection sampling trials that is comprised of 458 positive and 528 negative examples. The smallest loss implies that it is expected to pay 1.1% of the total loss caused by incorrect confidential access control

(i.e., misclassification). From the false positive perspective (zero false alarm), you do not worry at all about the leaking of confidential information. 39% false negative rate means that there would be 39 out of 1000 valid requests to the confidential information that are mistakenly rejected. This inconveniences employees because they have to access particular information for their projects, but the system does not authorize. This trend holds good for the remaining four categories.

The primary reason that makes costing effective is its ability to generate a sample that is comprised of nearly even number of positive and negative examples. For example, each resampled set for the "interest" category has only about 500 documents (actually ranging from 502 to 555). Ranging about 55% to 60% of the documents in each set are positive, even though on the original dataset it was only 3.6% (513 out of 14,128). Moreover it takes relatively less time to train the classifier with costing because sample set is far smaller than the original training example. However this property hindered the performance of SVM because its performance is sensitive to a skewed class distribution, even with regularization (i.e., assigning a higher cost factor, $C = 100$). In other words, it is difficult for a SVM to find the optimal hyperplane separating two classes if the size of one class is relatively smaller than the others. The result in table 2 confirmed this hypothesis in that the more training examples are available for SVM without wrappers, the less cost it is paid (e.g., from 32% (13967 out of 42625) for "livestock" category to 10% (108108 out of 1090498) for "earn" category).

Another reason, we believe, that the cost resulted in good performance is that our method for assigning cost distinguished well positive from negative examples. By assigning at least more than 100 times cost to positive examples, it helps the costing choose the more important examples as sample.

The metacost did not show a good performance because there might be overfitting caused by a random resampling with replacement. To avoid such overfitting, one might think a resampling without replacement where an instance, x, is drawn from a distribution and the next sample is drawn from the set $S - x$. However this approach also fails because it keeps the size of the original distribution smaller and eventually there is nothing left to be chosen.

5 Related Work

As many data mining techniques have been applied to various real-world application domains, the usefulness of the cost-sensitive learning drew attention from the public.

Lee and his colleagues [7] introduced a cost-sensitive framework for the intrusion detection domain and analyzed cost factors in detail. Particularly, they identify the major cost factors (e.g., costs for development, operation, damages and responding to intrusion) and then applied a rule induction learning technique (i.e., RIPPER) to this cost model, in order to maximize security while minimizing costs. However their cost model should be changed manually if a system's cost factors are changed.

Maloof [8] utilized two sampling methods, such as under- and over-sampling for efficient learning of skewed data set. The under/over sampling are stratification techniques that generate a set of samples from the original data according to a certain criteria [2]. In the under-sampling, all instances of the class j with highest $P'(j)$ are re-

tained, and a fraction $P'(i)/P'(j)$ of the examples of each other class i is chosen at random for inclusion in the resampled training set, where $P'(j) = C(j)P(j)/\sum_j C(j)P(j)$ and $C(j) = \sum_i C(i|j)$. $C(j)$ is the cost of misclassifying an instance of class j, irrespective of the class predicted. In the over-sampling, on the contrary, all examples of the class j with lowest $P'(j)$ are retained, and then the examples of every other class i are duplicated approximately $P'(i)/P'(j)$ times in the training set. Both sampling methods have drawbacks; the under-sampling reduces the amount of data available, which might cause the increase of cost whereas the over-sampling avoids the loss of training data but may significantly increase learning time. Moreover these techniques are only applicable when there is a slight difference in the class frequency. In our case, these two techniques resulted in the following distribution of five selected topics; *under-sampling*: livestock (114 → 0.35), corn (253 → 3.23), interest (513 → 5.46), acq (2448 → 1278.68), earn (3987 → 3987) and *over-sampling*: livestock (114 → 9844.61), corn (253 → 89809), interest (513 → 617137), acq (2448 → 35452826), earn (3987 → 110543424). Since there are too few examples available for training by under-sampling whereas there are too many examples to computationally process by over-sampling, we did not apply those two methods to our scenario.

Fan and his colleagues [16] proposed a new method called "AdaCost" for reducing misclassification cost using boosting. In particular, the idea is to take an unequal care for examples according to their cost while learning rule – by assigning high initial weights to costly weights and by updating rule in taking cost into account. Their approach is similar to the costing in that the performance is improved by averaging (i.e., weighted bagging vs bagging). The finding in comparison of AdaCost with AdaBoost by "Chase credit card data " is quite similar to ours – AdaBoost reduces misclassification error significantly but does not reduce cost for misclassifying.

6 Conclusion and Future Work

In this paper we test the effectiveness of cost-sensitive learning for confidential access control. The goal of this work is to develop a reliable process for confidential access control based on the need-to-know principle; the request for access to a unit of confidential information is accepted only if the content of the requested item is relevant to the requester's task. We model such a dichotomous decision (i.e., to reject or accept the request) in a machine learning framework. A false positive occurs when the system accepts a request that should not have been accepted whereas a false negative occurs when the system rejects a request that should have been accepted. For both errors, the system pays the cost for misclassification. From the security perspective, the cost for a false positive is more expensive than that of false negative because the former is a serious security problem because confidential information, which should not be revealed, can be accessed.

In order to achieve our goal we need to find a classifier that minimizes the cost and false positive rate while holding false negative rate reasonable row. We utilized two wrappers for cost-sensitive learning because the best-performing error-minimizing classifiers do not concern unequal cost for misclassification. From the comparison of the cost-sensitive learners with the error-minimizing classifiers, we found that costing

showed the best performance. In particular, it requires far less training data for much better results, in terms of the smallest cost paid, the lowest false positive rate, and the lower false negative rate. The benefit of smaller training data is two-fold; First, obviously it takes less time to train the classifier; Second, it enables a human administrator to conveniently identify arbitrary subsets of confidential information, in order to train the initial classifier.

Since we found our metric for cost assignment useful, as future work, we would like to generalize this idea. In this work, we primarily focused on testing this framework for the text domain. We would like to investigate the usefulness of this approach in different type of media, such as image, video, etc. Although to our knowledge, the machine learning approach is a novel one for access control, it would be very interesting if we compare the effectiveness of our framework with conventional document management systems (e.g., ACL-based systems).

7 Acknowledgement

This research has been supported by ARDA under the CTA sub-contract and by DARPA grant F30602-03-C-0009. We would like to thank Robin Glinton and Sean Owens for their friendly and responsive help and comments.

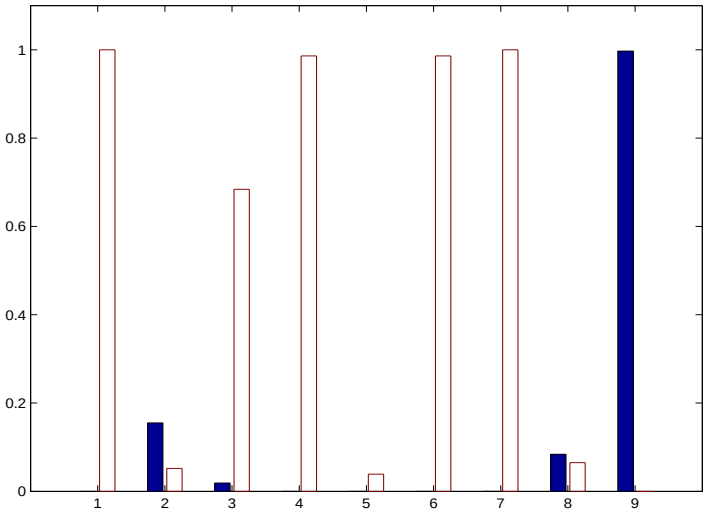

Figure 3: A pair of of false positive (filled bar) and false negative (empty bar) for "corn" category is presented.

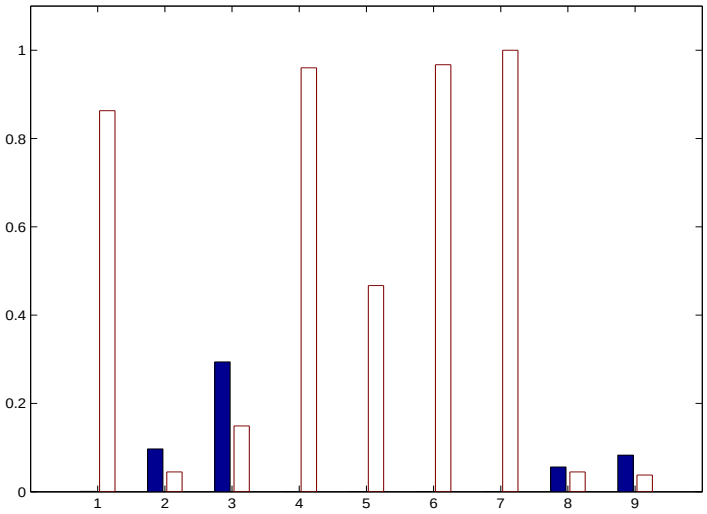

Figure 4: A pair of of false positive (filled bar) and false negative (empty bar) for "interest" category is presented.

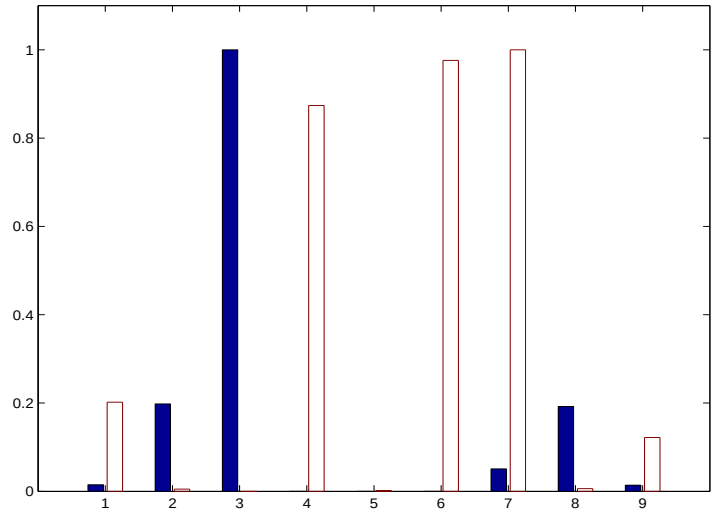

Figure 5: A pair of of false positive (filled bar) and false negative (empty bar) for “acq” category is presented.

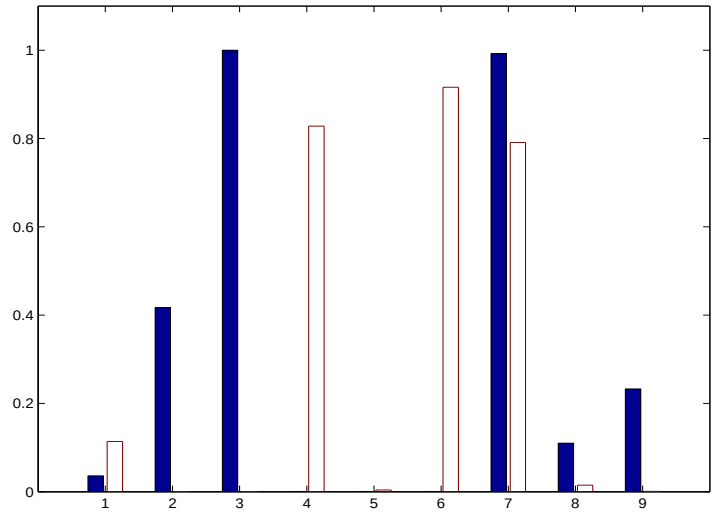

Figure 6: A pair of of false positive (filled bar) and false negative (empty bar) for “earn” category is presented.

References

[1] D. Chris and H. R. C. Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of International Conference on Machine Learning*, pages 239–246, 2000.

[2] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[3] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[4] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Lab Palo Alto, 2003.

[5] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European Conference on Machine Learning*, 1998.

[6] F. Keinosuke. *Introduction To Statistical Pattern Recognition*. Morgan Kaufmann, 1990.

[7] W. Lee, M. Miller, S. Stolfo, K. Jallad, C. Park, E. Zadok, and V. Prabhakar. Toward cost-sensitive modeling for intrusion detection. *ACM Journal of Computer Society*, 10:5–22, 2002.

[8] M. A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 Workshop on Learning from Imbalanced Data Sets*, 2003.

[9] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[10] D. R. O., H. P. E., and S. D. G. *Pattern Classification*. Wiley-Interscience, 2001.

[11] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 2004.

[12] H. Schutze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of International ACM Conference on Research and Development in Information Retrieval*, pages 229–237, 1995.

[13] Y.-W. Seo, J. Giampapa, and K. Sycara. A multi-agent system for enforcing need-to-know security policies. In *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems (AAMAS) Workshop on Agent Oriented Information Systems (AOIS)*, pages 163–179, 2004.

[14] K. Torkkola. Linear discriminant analysis in document classification. In *IEEE Workshop on TextMining*, 2001.

[15] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[16] F. Wei, S. S. J., Z. Junxin, and C. P. K. Adacost: Misclassification cost-sensitive boosting. In *Proceedings of International Conference on Machine Learning*, pages 97–105, 1999.

[17] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[18] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of International Conference on Machine Learning*, pages 412–420, 1997.

[19] B. Zadrozny, J. Langford, and N. Abe. A simple method for cost-sensitive learning. Technical report, IBM Tech Report, 2002.