LLNL-CONF-407542

Building Ultra-Low False Alarm Rate Support Vector Classifier Ensembles Using Random Subspaces

B. Y. Chen, T. D. Lemmond, W. G. Hanley

October 7, 2008

2009 Symposium on Computational Intelligence and Data Mining
Nashville, TN, United States
March 30, 2009 through April 2, 2009

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Building Ultra-Low False Alarm Rate Support Vector Classifier Ensembles Using Random Subspaces

Barry Y. Chen, Tracy D. Lemmond, and William G. Hanley

Abstract — **This paper presents the Cost-Sensitive Random Subspace Support Vector Classifier (CS-RS-SVC), a new learning algorithm that combines random subspace sampling and bagging with Cost-Sensitive Support Vector Classifiers to more effectively address detection applications burdened by unequal misclassification requirements. When compared to its conventional, non-cost-sensitive counterpart on a two–class signal detection application, random subspace sampling is shown to very effectively leverage the additional flexibility offered by the Cost-Sensitive Support Vector Classifier, yielding a more than four-fold increase in the detection rate at a false alarm rate (FAR) of zero. Moreover, the CS-RS-SVC is shown to be fairly robust to constraints on the feature subspace dimensionality, enabling reductions in computation time of up to 82% with minimal performance degradation.**

I. INTRODUCTION

In many two-class detection applications of practical significance, the two types of classification error, missed detections and false alarms, are associated with inherently unequal costs. Whether these costs are tangible (e.g., loss of money, life or time), or intangible (e.g., loss of security or opportunity), they generally impose explicit requirements that must be effectively addressed via appropriate models and methodologies. This paper introduces a novel classification methodology called the *Cost-Sensitive Random Subspace Support Vector Classifier* (CS-RS-SVC) that has been developed specifically to address these types of applications.

The CS-RS-SVC is an ensemble-based methodology that utilizes the Cost-Sensitive Support Vector Classifier (CS-SVC) [1, 2] as its base classifier. Conventional (i.e., non-cost-sensitive) SVCs, like many other classifiers, automatically learn decision boundaries in feature space that separate two classes of interest and minimize the *overall* error. In this paradigm, cost-sensitivity is often emulated by selecting a decision threshold such that a specified error bound is not exceeded, effectively translating the boundary in feature space. Not surprisingly, translation of the decision boundary often results in an unacceptable increase in the less egregious error type. Cost-sensitive classifiers, such as the CS-SVC, cost-sensitive Multi-layer Perceptrons [3], and cost-sensitive decision trees [4], on the other hand, *transform* this boundary to optimally account for unequal error costs.

Standalone classifiers such as those described above can be greatly enhanced by learning numerous instances of each and combining their decisions. These multi-classifier systems, or classifier *ensembles*, almost always achieve performance superior to that of their individual components. One of the most effective of these leverages the concept of Bootstrap Aggregation (i.e., *bagging*) [5], in which each base classifier is trained on a bootstrapped sample of the original training set. The overall ensemble class decision is determined by voting the base classifier decisions. Valentini and Dietterich demonstrated the effectiveness of bagging low-bias SVCs in [6].

A further enhancement to the bagging approach is the Random Subspace methodology developed by Ho [7], which introduces additional diversity via random sampling with replacement of the feature subspaces. The Random Subspace method (i.e., bagging + random feature subspace sampling) forms the basis for the highly effective classifier formalized by Breiman called the *Random Forest* [8].

The new CS-RS-SVC methodology presented in this paper leverages both of these ensemble enhancements, but in contrast to the cost-sensitive ensembles developed in [9, 10, and 11], it derives its cost-sensitivity strictly from CS-SVC base classifiers. This research demonstrates, via a two-class detection problem, that the CS-RS-SVC achieves significantly higher detection rates at lower false alarm rates than comparable non-cost-sensitive systems. The paper is organized as follows: Sections II and III will introduce the conventional and cost-sensitive SVCs, respectively, and Section IV will provide a detailed description of the CS-RS-SVC learning algorithm. In Section V, we will compare the detection performance of the CS-RS-SVC to conventional SVC ensembles when applied to a two-class signal detection problem with high-cost false alarms. Our conclusions are summarized in Section VI.

II. CONVENTIONAL SUPPORT VECTOR CLASSIFIER

A. Overview

A Support Vector Classifier (SVC) (a.k.a. Support Vector Machine) [12, 13, 14] is a classification algorithm that combines two powerful methods - the maximal margin classifier and the kernel trick. A maximal margin classifier is the hyperplane that best separates two classes of data while maximizing the distance between the hyperplane and each data sample. This hyperplane can be determined via optimization of an equation that can be expressed as an inner product between pairs of training samples. These inner products, through substitution of kernel functions in the optimization equations (i.e., "the kernel trick"), are computed in a much higher dimensional feature space, resulting in nonlinear maximal margin hypercurves.

More formally, given a set of training samples ($\mathbf{x}_i$, y_i), i = 1, …, m where $\mathbf{x}_i \in \Re^n$ and $y_i \in \{-1,+1\}$, the goal is to determine the hyperplane defined by the coefficients

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

B. Chen*, T. Lemmond, and W. Hanley are with Lawrence Livermore National Laboratory, Livermore, CA 94551 USA (*phone: 925-423-9429; fax: 925-422-8277; e-mail: {chen52, lemmond1, hanley3}@ llnl.gov).

$\mathbf{w} \in \Re^n$ and a scalar bias, b, that optimizes the following expression (1),

$$\begin{aligned}
&\underset{\mathbf{w},b,\xi,\rho}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_i \xi_i, \\
&\text{subject to} \\
&\qquad \xi_i \geq 0, \forall\ i = 1,\ldots,m \\
&\qquad y_i\big(k(\mathbf{x}_i,\mathbf{w}) + b\big) \geq \rho - \xi_i, \forall\ i = 1,\ldots,m \\
&\qquad \rho \geq 0
\end{aligned} \tag{1}$$

where ν is a user-specified regularization term that controls the curviness of the resulting hypercurve, $k(\mathbf{x}_i,\mathbf{x}_j)$ is the kernel function $k(\mathbf{x}_i,\mathbf{x}_j) \equiv \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, and $\phi(\cdot)$ is a mapping from input space to a potentially infinite feature space that encourages greater class separability. In the experiments presented in Section V, we use Gaussian kernel functions given by:

$$k(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2} \right) \tag{2}$$

where σ is the user-specified kernel width parameter that controls the sharpness of the resulting hypercurve.

Because the two classes are generally not completely separable, (1) contains an error penalization term given by the average ξ_i, a slack variable that measures the distance from the separating hyperplane to a wrongly classified $\mathbf{x}_i$. Note that this penalization term makes no distinction between error types. In Section III, we will discuss enhancements that leverage these slack variables to create the Cost-Sensitive SVC.

B. SVC Parameter Selection

Successfully applying an SVC to a particular problem of interest depends heavily upon the selection of its training parameters. For the conventional SVC with a Gaussian kernel, this involves selecting ν and σ so as to optimize a desired performance metric. Our preferred metric is based upon the Receiver Operating Characteristic (ROC) curve [15], which plots detection rates against a range of false alarm rates (FAR), providing a natural means for visualizing the tradeoffs between the two error types. The detection application on which we focus our attention (i.e., the "Hidden Signal Detection" application described in Section V), like many other real-world problems, requires extremely low false alarm rates. In order to more effectively assess performance in these extreme regions, we use the summary metric given by the area under the ROC curve (AUC) [15]. From a practical standpoint, the AUC over a false alarm interval of interest provides a measure of how well the classifier *discriminates* between two classes within the corresponding region. For SVC parameter optimization, we utilize the AUC over the FAR interval [0, 10^{-3}], which we denote AUC10^{-3}.

The brute-force methodology for optimal parameter selection with respect to the AUC10^{-3} requires successive grid searches of increasing resolution over all feasible SVC parameters and selecting those that yield the highest 5-fold AUC10^{-3} estimate (i.e., an unbiased estimate of AUC10^{-3} via *n*-fold cross-validation [16]). Computationally, this is quite costly, as each grid point necessitates the training and testing of five SVCs. To reduce the computational complexity, we employ the Nelder-Mead (Simplex or Amoeba) optimization algorithm [17]. In our experience, the Nelder-Mead algorithm reduces the number of grid points that require processing by up to a factor of ten and converges to the same parameter settings. However, one can trade off optimality for speed, if necessary, by setting a higher tolerance for convergence of the algorithm.

III. Cost-Sensitive Support Vector Classifier

Unlike the conventional SVC, the Cost-Sensitive Support Vector Classifier is designed to perform more effective classification under unequal error conditions. Originally developed by Chew, et. al. [1], and later re-parameterized by Davenport, et. al. [2], the CS-SVC allows the user to specify unequal penalties for false alarms and missed detections. The CS-SVC maximal margin optimization equation as defined in [1] is given by (3).

$$\begin{aligned}
&\underset{\mathbf{w},b,\xi,\rho}{\text{minimize}} \ \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{\gamma}{m}\sum_{i:y_i=+1} \xi_i + \frac{1-\gamma}{m}\sum_{i:y_i=-1} \xi_i, \\
&\text{subject to} \\
&\qquad \xi_i \geq 0, \forall\ i = 1,\ldots,m \\
&\qquad y_i\big(k(\mathbf{x}_i,\mathbf{w}) + b\big) \geq \rho - \xi_i, \forall\ i = 1,\ldots,m \\
&\qquad \rho \geq 0
\end{aligned} \tag{3}$$

Note that the fundamental distinction between the SVC optimization equations given by (1) and those of the CS-SVC lies exclusively in the treatment of the slack variable penalty term, which has been split into two separate weighted sums: one for errors on the positive samples and another for negative samples. Thus, by setting γ to a value other than 0.5, the penalties for false alarms and missed detections can be given unequal weighting. In the CS-SVC parameterization, the user is responsible for specifying the training parameters ν and γ, along with any kernel parameters (e.g., σ for the Gaussian kernel). An equivalent re-parameterization of the CS-SVC is described in [2] that replaces ν and γ with $\nu+$ and $\nu-$, which are shown to be the respective upper bounds on the positive and negative class margins for the training set. For example, when $\nu- < \nu+$, the optimization procedure results in maximal margin hypercurves that preferentially minimize the false alarm rate. It is this flexibility that often enables CS-SVCs to outperform conventional SVCs at ultra-low false alarm or missed detection rates.

IV. SVC Ensembles with Bagging and Random Subspace Sampling

Inspired by the work of Ho [7] and Breiman [8] whose ensemble classifier learning algorithms combine bagging with the random sampling of feature subspaces, we developed similar ensemble systems incorporating the CS-SVC as the base classifier (CS-RS-SVC) and contrasted these with conventional SVC ensembles (RS-SVC). Fig. 1 shows the pseudo-code for training either RS-SVC or CS-RS-SVC systems with N base classifiers on a data set $\boldsymbol{D}$ of size m.

The algorithm consists of two phases. Since each base classifier will be trained on a (not necessarily unique) subspace of the feature set, the first phase involves pre-computing the optimal training parameters for each of the subspaces included in what we call the *feature subspace sampling pool (fss_pool)*. Often, a user may choose to populate this pool with all possible feature combinations (i.e., 2^F-1 subsets of F features). However, when computational efficiency is a significant concern, the member subspaces may be *dimensionally constrained*, a modification whose strengths and weaknesses are discussed in detail in Subsection V-*D*.

In the second phase of the algorithm, each base classifier in the ensemble is trained on a bootstrapped training set and a feature subspace randomly sampled from the feature subspace sampling pool. Each sample in the bootstrapped training set is projected onto the selected feature subspace, and this projected training set, along with the optimal subspace parameters, is used to train the corresponding base classifier. Like typical bagged ensemble classifiers, the resulting ensemble produces a class prediction for a new test sample via voting of the individual base classifiers' decisions.

It is imperative to recognize that the parameter estimates computed in the first stage of the algorithm are necessarily optimized for the entire training set. Consequently, they are at best loosely optimal for the bootstrapped data sets used to train the base classifiers in the second stage. An alternative implementation of the RS-SVC and CS-RS-SVC algorithms involves computing the optimal parameters on the fly for each of the bootstrapped training sets. This approach would yield better parameter estimates and would prove to be computationally more efficient when the number of feature subspaces is large relative to the ensemble size. However, the appropriate choice is highly dependent upon the application and its efficiency requirements. In the Hidden Signal Detection application presented in the following section, the ensemble size greatly exceeded the cardinality of the feature subspace sampling pool, motivating our choice to pre-compute the SVC and CS-SVC parameters.

```
(Phase I) Compute_Optimal_Parameters:
foreach $subspace_i$ in $fss\_pool$
    $D_i'$ ← project($D$, $subspace_i$)
    $OptPars_i$ ← optimize_SVC_params($D_i'$)
end

(Phase II) Train_Ensemble:
for i=1 to $N$
    $D_i$ ← sampleWithReplacement($D$, $m$)
    $subspace_i$ ← sampleOne($fss\_pool$)
    $D_i'$ ← project($D_i$, $subspace_i$)
    $SVCBaseClassifer_i$ ← TrainSVC($D_i'$, $OptPars_i$)
end
```

Fig. 1 – Pseudo-code for training RS-SVC and CS-RS-SVC ensembles.

V. Comparing Classifiers on The Hidden Signal Detection Application

The Hidden Signal Detection application is a two-class problem whose goal is to detect an embedded signal in a data sample. We computed a total of eight real-valued features useful for detecting the presence of an embedded signal, each of which was normalized by subtracting its mean and dividing by its standard deviation. For this application, a detection rate greater than 50% is desired, but since each detection event requires a considerable amount of costly post-hoc analysis, false alarms must be minimized.

The Hidden Signal dataset consists of a training set and a separate test set. The training set consists of 7,931 clean samples (negative class) and 7,869 samples with an embedded signal (positive class). The test set contains 9,426 positive and 179,528 negative samples, a class size which allows us to evaluate classifier performance at the desired ultra-low false alarm rates (i.e., nonzero FAR values as low as $5.57\text{x}10^{-6}$).

All of the ensembles in the following experiments consist of 500 base classifiers, a size empirically determined to be sufficient for performance to plateau. In each case, we determined the optimal parameter settings using a convergence tolerance of 1% and setting the maximum number of simplex iterations to 50. The initial parameters we used for the SVCs were $\nu = 0.1$ and $\sigma = 1$, with step sizes of 0.075 and 0.5, respectively. Similarly, the initial parameters used for the CS-SVCs were $\nu+ = \nu- = 0.1$ and $\sigma = 1$, with step sizes of 0.075, 0.075, and 0.5, respectively.

For all of our experiments, we modified and leveraged the LIBSVM software package [18] for the training and testing of SVCs. Our modified LIBSVM contains a wrapper function to implement the Nelder-Mead optimization of SVC training parameters as well as the CS-SVC modifications specified by [2].

In the following discussions of performance, we will present our results in a form that allows us to thoroughly examine detection rates over the low false alarm regions of interest. Hence, the FAR axis of the ROC curves are plotted on a log scale, and in some cases we have clipped the region to enable more detailed visualization. In an effort to assess the statistical significance of our ensemble performance, we trained and tested eleven ensembles for each methodology, varying only the seed of the random number generator. For each performance metric, we then computed the median value along with the 10th and 90th percentiles for each point on the curve in a manner consistent with the "vertical averaging" approach described in [15].

A. Conventional SVC Ensembles

We begin our performance analysis by demonstrating the respective advantages of the bagging and Random Subspace ensemble methodologies in the low false alarm regions of interest. Specifically, we wish to compare the performance of these ensemble classifiers to that of a singleton SVC. ROC curves indicating the performance on the test set for all three of these methods are shown in Fig. 2.

It is immediately apparent from Fig. 2 that for FARs less than 5x10^{-4}, the Random Subspace method appears to yield significantly higher detection rates than both the singleton SVC and the bagged ensemble. This is a trend that we will observe repeatedly throughout this discussion. The fundamental reason for this behavior is related to the greater diversity among the base classifiers that is afforded via feature subspace sampling, a characteristic that is discussed at length in [8], and one that we will briefly explore in the next subsection. The ensemble that was created via bagging alone also appears to significantly improve upon the singleton classifier, increasing the detection rate by up to 20 percentage points. However, this improvement is far less pronounced than that of the more diverse Random Subspace ensemble.

Note that the distance between the percentile bands for each of the ensemble classifiers clearly reflects its base classifier diversity. That of the bagged ensemble is extremely narrow (almost invisible upon visual inspection), indicating very little performance variability over the eleven runs. Hence, it is not surprising to see that its performance trend (i.e., the *shape* of the ROC curve) strongly resembles that of the singleton SVC.

Although the Random Subspace method appears to be less effective at higher false alarm rates, it clearly achieves our cost-sensitivity requirements over the more extreme regions. Its detection rate at the lowest nonzero FAR exceeded that of the other classifiers by a factor of more than five. Additionally, it is the only classifier to achieve a nonzero detection rate (i.e., 9.4% - not shown) at a FAR of zero.

In keeping with our discussion of the AUC in Section II, we also compared the classification performance of these classifiers via the normalized AUC over the ultra-low FAR intervals [0,10^{-5}], [0,10^{-4}] and [0,10^{-3}]. Box and whisker plots (Fig. 3) show the 25th, 50th, and 75th percentiles (box) along with the smallest and largest non-outlier AUCs (whiskers) over each of these ranges. The separation of these boxes over all of the regions provides further evidence that the RS-SVC achieves superior performance at low FARs.

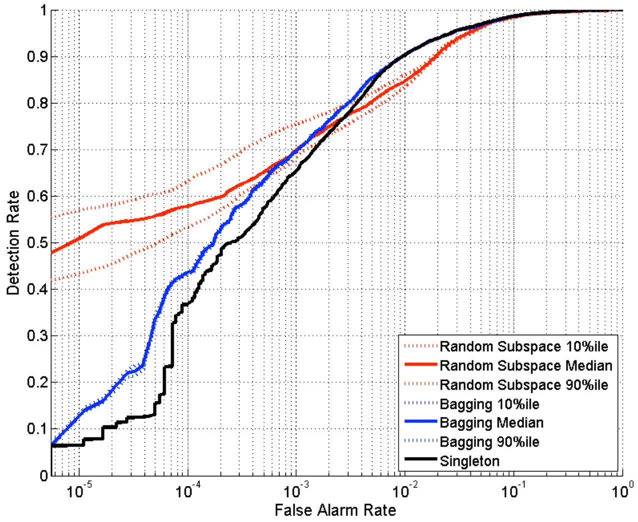

Fig. 2 – ROC curves of conventional SVC singleton, bagging, and Random Subspace systems.

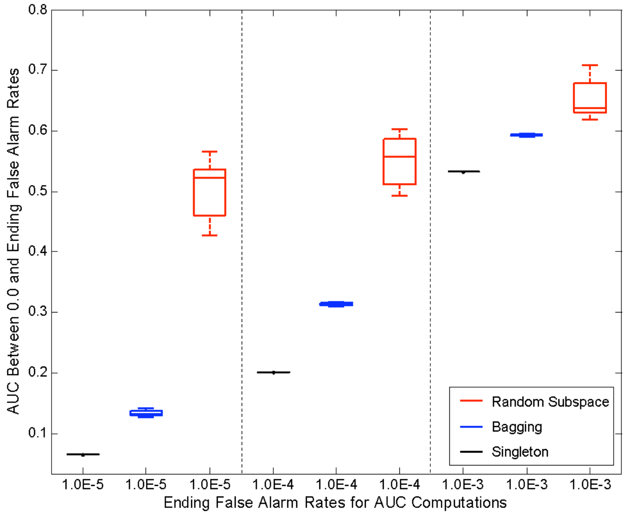

Fig. 3 – Box plots of AUC computed over the FAR range [0.0, *ending FAR*] for conventional SVC systems.

In most practical applications, we are not only interested in quantifying the error rates, but also the expected cost incurred by a classification system. This metric is given by

$$EC = p(+) \cdot (1 - DR) \cdot c(miss) + p(\text{-}) \cdot FAR \cdot c(falsealarm) \quad (4)$$

where DR is the detection rate, $p(\cdot)$ is the prior probability for each class, and $c(\cdot)$ is the cost for each type of error. To enable visualization of this metric, Drummond and Holte developed "cost curves" that express expected cost as a function of the class priors and costs [19]. Specifically, cost curves plot the expected cost (normalized by its maximum value) versus the probability cost function (PCF), which is given by:

$$PCF = \frac{p(+) \cdot c(miss)}{p(+) \cdot c(miss) + p(-) \cdot c(falsealarm)}. \quad (5)$$

Assuming equal priors, PCF is small when the cost for false alarms is large relative to that of missed detections. In the

Hidden Signal Detection application, the cost of a false alarm is considered to be at least 100 times more costly than a missed detection, making classifiers whose cost curves are lower for *PCF* < 0.01 more desirable. Fig. 4 shows cost curves for the conventional singleton, bagged, and Random Subspace classifiers. The bagged and Random Subspace ensembles appear to attain significantly lower expected costs than the singleton SVC over the range of interest. For values of the *PCF* < 0.005, the Random Subspace classifier is heavily favored.

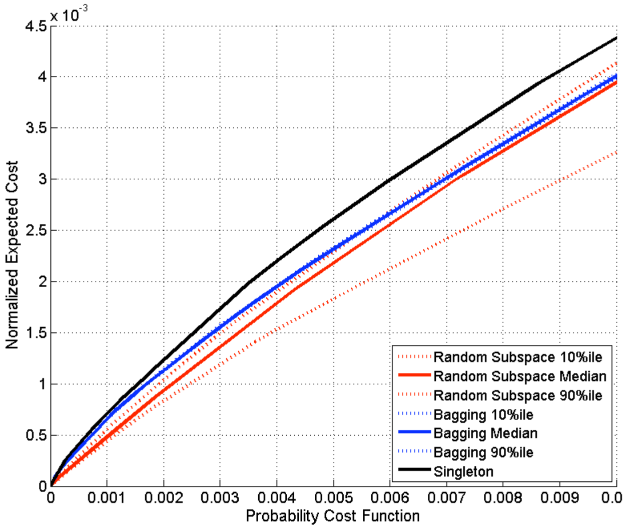

Fig. 4 – Cost curves of conventional SVC systems focused on PCF regions where false alarms are more than one hundred times more costly.

B. *Cost-Sensitive SVC Systems*

Having established the advantages of bagging and feature subspace sampling in conventional SVC ensembles, we wish to demonstrate the additional gains afforded by our cost-sensitive variants of these methods. We repeated the experiments in the previous subsection utilizing the Cost-Sensitive SVCs, as described in Section III, as the base classifiers for the ensembles. The ROC curves for these cost-sensitive systems are shown in Fig. 5. Like the conventional bagged ensemble, the cost-sensitive variant significantly outperforms its corresponding singleton, and their ROC curves are similar in shape. Unlike our previous experiment, however, the cost-sensitive Random Subspace ensemble does not significantly underperform the other classifiers at any FAR. In fact, at 5.57x10^{-6} FAR, the median detection rate for the CS-RS-SVC is more than six times that of the other classifiers, and at a FAR of zero, it attains a 38.8% median detection rate (not shown) as compared to zero for both the bagged and singleton CS-SVCs.

The AUC box and whisker plots in Fig. 6 indicate similar behavior as in the conventional case, but the advantage enjoyed by the CS-RS-SVC is far more pronounced, with relatively high AUCs even across the lowest FAR region.

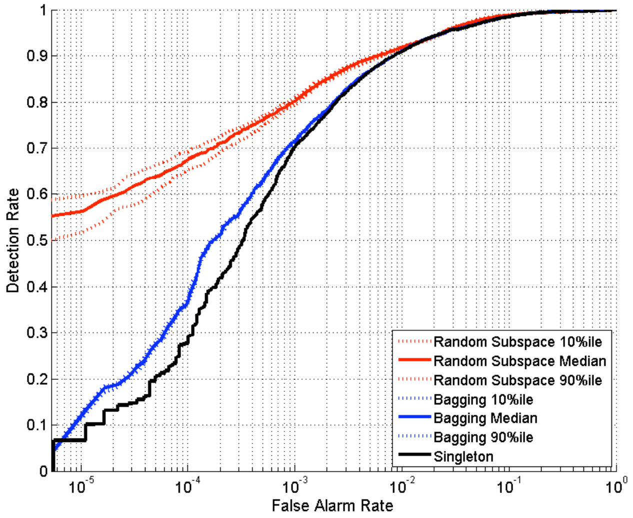

Fig. 5 – ROC curves of cost-sensitive SVC singleton, bagging, and Random Subspace systems.

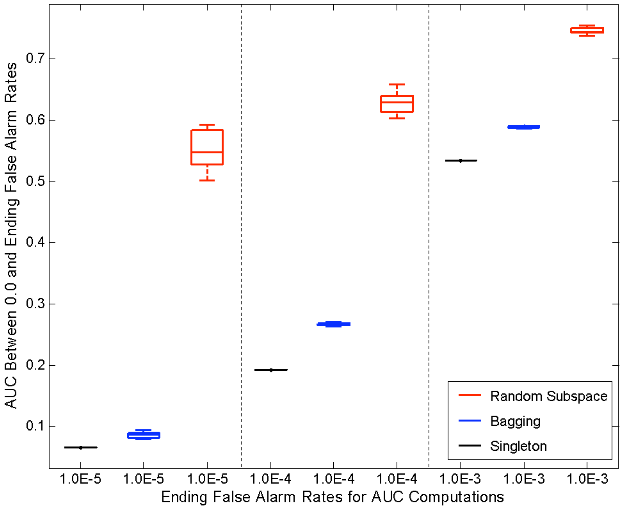

Fig. 6 – Box plots of AUC computed over the FAR range [0.0, *ending FAR*] for cost-sensitive SVC systems.

Additionally, the cost curves for the cost-sensitive classifiers, shown in Fig. 7, suggest significant reductions in expected cost in the low *PCF* region of interest when progressing from singleton to bagging, and an even more significant reduction when Random Subspace Sampling is employed.

Interestingly, a comparison of these results to those in the prior subsection suggests that the performance gain due to Random Subspace Sampling versus that due to bagging alone appears to be much more pronounced for the cost-sensitive classifier variants. This observation strongly suggests that the Random Subspace ensemble is able to leverage the flexibility of the CS-SVC far more effectively than bagging alone. Specifically, the cost-sensitivity of the base classifiers appears to combine synergistically with the multi-dimensional slices through feature space to *amplify* ensemble classification performance.

We know from [8] that ensemble performance is enhanced by both higher strength (i.e., accuracy) and lower correlation of its base classifiers. Figures 3 and 6 indicate that, for this data set, the strengths of the standalone conventional and cost-sensitive SVCs are roughly equivalent. However, because the CS-SVC incorporates an additional parameter, it is a more flexible model than the standard SVC, and hence, is capable of achieving greater variability. A reasonable conjecture, then, is that the observed amplified performance is at least partially derived from reduced correlation among the CS-SVC base classifiers under feature subspace sampling. In fact, $\bar{\rho}_{CostSens} = .403$ and $\bar{\rho}_{Conv.} = .427$, computed according to [8].

Though the above argument is reasonable, we suspect that the underlying cause for this phenomenon is nontrivial and arises from multiple factors that depend upon both the models and the data. Further study would be necessary to fully investigate these interactions.

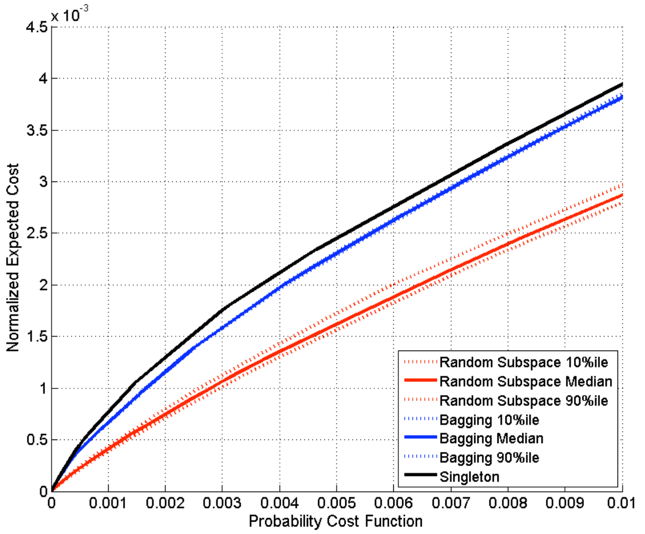

Fig. 7 – Cost curves of cost-sensitive SVC systems focused on PCF regions where false alarms are more than one hundred times more costly.

C. Conventional Versus Cost-Sensitive Random Subspace Systems

In the prior subsections, we have explored the effect of each ensemble methodology on classifier performance, and found that the Random Subspace approach yields the greatest benefits. In this subsection, we contrast the performance of these classifiers with respect to the cost-sensitivity of their base SVC methodology. The ROC curves plotted in Fig. 8 clearly show that the median detection performance of the CS-RS-SVC exceeds that of the RS-SVC over the entire FAR range. Furthermore, for FARs within the interval [5x10^{-5}, 8x10^{-2}] their respective percentile bands do not overlap, providing strong evidence that CS-RS-SVC significantly outperforms the RS-SVC over this region. Although the percentile bands do overlap for FARs < 5x10^{-5}, the CS-RS-SVC still appears to maintain a slight advantage. It is interesting to note that the CS-RS-SVC percentile bands are even narrower than those of its conventional counterpart, suggesting greater performance stability in low FAR regions.

A comparison of the AUC box and whisker plots for these ensembles, shown in Fig. 9, further supports the behaviors observed in Fig. 8. The CS-RS-SVC appears to achieve significantly higher detection performance for FAR values up to 10^{-4}. In the most extreme FAR region, the interquartile

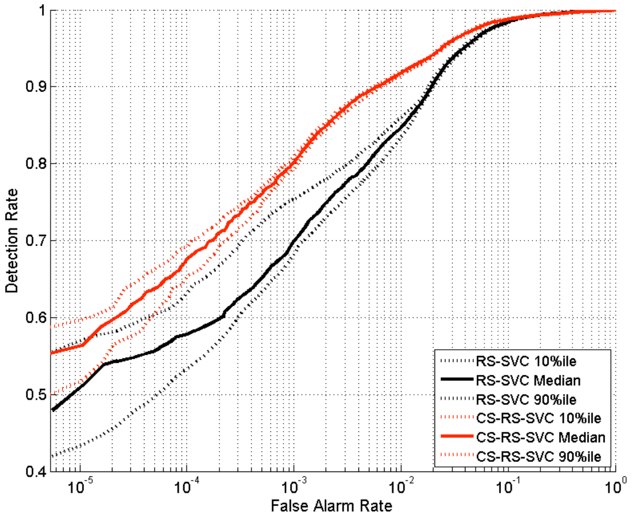

Fig. 8 – ROC curves of conventional (RS-SVC) and cost-sensitive Random Subspace (CS-RS-SVC) classifiers.

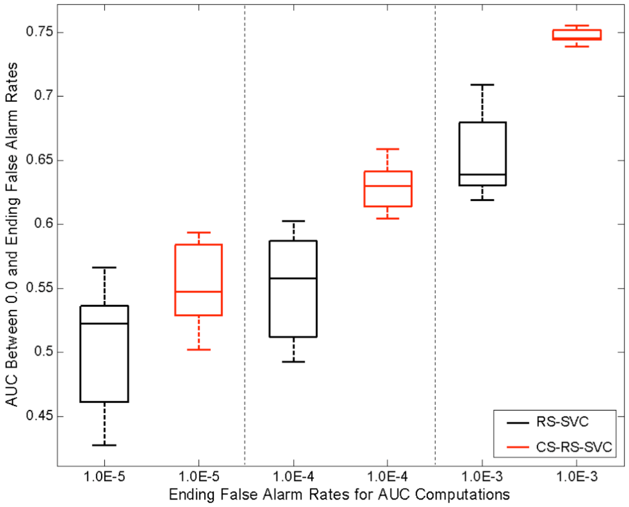

Fig. 9 – Box plots of AUC computed over the FAR range [0.0, *ending FAR*] for cost-sensitive SVC systems.

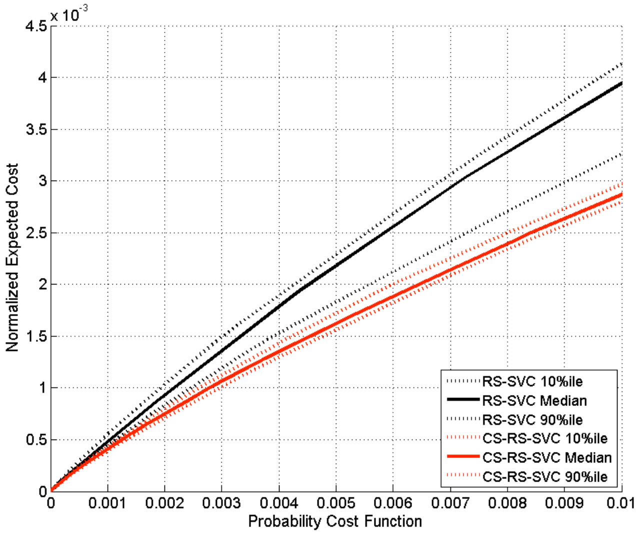

Fig. 10 – Cost curves of conventional (RS-SVC) and cost-sensitive Random Subspace (CS-RS-SVC) systems focused on the low PCF region of interest.

ranges of their AUCs overlap slightly, but the median AUC of the CS-RS-SVC still lies above the interquartile range of the RS-SVC's AUC, providing further evidence, albeit weaker, that the CS-RS-SVC is more effective than the RS-SVC. Finally, Fig. 10 suggests that the CS-RS-SVC incurs significantly lower expected cost than the RS-SVC over the *PCF* region of interest where false alarms are at least 100 times more costly than missed detections.

Again, the CS-RS-SVC's percentile bands are much tighter than those of the RS-SVC, indicating greater performance stability and reliability of the detection system in practical applications.

D. Constraining Subspace Dimensionality for Random Subspace Systems

In Section IV, we encountered the issue of computational complexity and proposed the possible use of constrained dimensionality. Specifically, we wish to explore the effects of constraining the feature subspaces in the sampling pool to *only* those of dimensionality *d*, as compared to unconstrained sampling over all feature subspaces. Fig. 11 shows ROC curves for RS-SVC systems with feature subspaces constrained to dimensionalities *d* = 1, 2, and 3, along with the RS-SVC using unconstrained sampling. The performance increases dramatically from *d* = 1 to *d* = 2, where it appears to achieve its peak. For *d* = 3, the performance degrades and continues to decline for *d* > 3 (not shown, to preserve clarity). When compared to the unconstrained RS-SVC, the *d* = 2 case appears to enjoy a slight advantage, and also has tighter percentile bands. A similar plot for the CS-RS-SVC (Fig. 12) indicates that *d* = 2 is at least as effective as the unconstrained ensemble.

To investigate the mechanisms underlying this phenomenon, we revisited the concepts of base classifier strength and correlation. Specifically, [8] shows that higher strength and lower correlation among base classifiers yield better ensemble performance. Since our application demands low false alarm rates, we plotted these metrics as a function of dimension for each ensemble with respect to the negative class to gain greater insight into their behavior (Fig. 13).

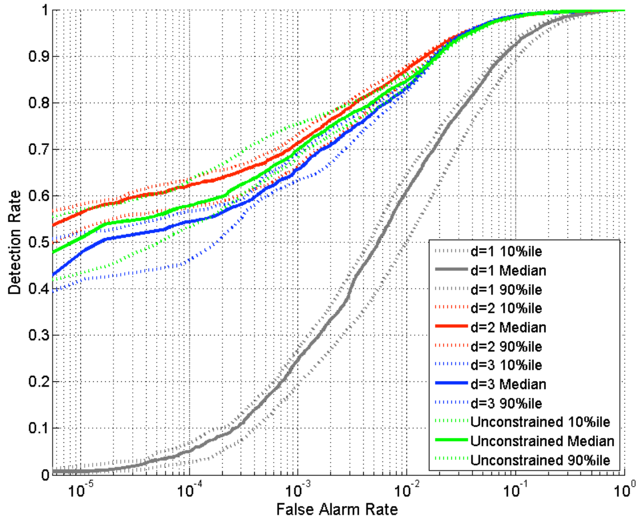

Fig. 11 – ROC curves of conventional Random Subspace ensembles with unconstrained and dimensionality-constrained feature subspaces.

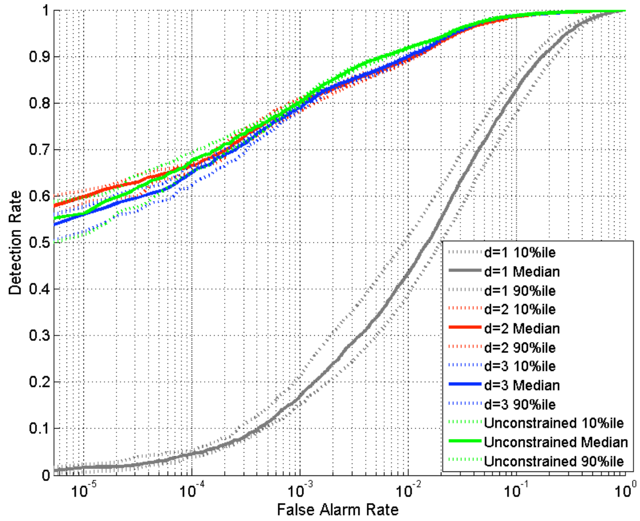

Fig. 12 – ROC curves of cost-sensitive Random Subspace ensembles with unconstrained and dimensionality-constrained feature subspaces.

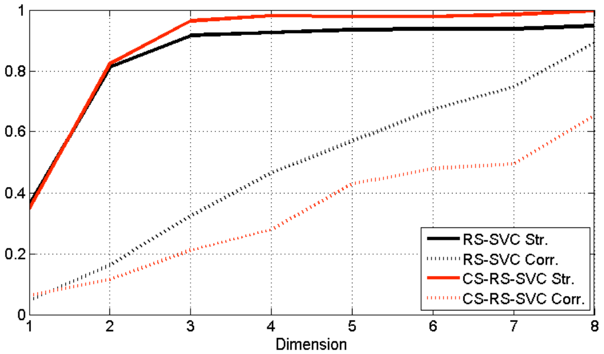

Fig. 13 – Strength and correlation of the base classifiers on negative samples for RS-SVC and CS-RS-SVC versus dimension constraint, *d*.

Note that, in general, the CS-RS-SVC consists of stronger and more diverse (i.e., less correlated) base classifiers than the RS-SVC with respect to the negative class. In each case, the strength is extremely poor at *d* = 1 and rises dramatically at *d* = 2, overwhelming the change in correlation and leading to a tremendous performance gain. However, as *d* continues to increase, the rate of correlation change for the RS-SVC exceeds that of strength, negatively impacting performance at higher dimensions. In contrast, the CS-RS-SVC maintains roughly the same rate of change in both strength and correlation as *d* increases from 2 to 3, yielding statistically similar performance for these dimensions. These observations are consistent with the ROC curves in Figs. 11 and 12. Remarkably, at *d* = 8, where the ensembles are identical to bagging alone, the correlation among the CS-RS-SVC base classifiers with respect to the negative class remains dramatically lower than that of the RS-SVC. At first glance, this appears to contradict our findings of Subsection V-*B* that bagged ensembles are not greatly enhanced when their base classifiers are made cost-sensitive. To clarify this result, we computed the correlation and strength values at *d* = 8 over *both classes* for each ensemble classifier. These values were found to be nearly identical (i.e., $\overline{\rho}_{CostSens} = .915$, $s_{CostSens} = .946$, $\overline{\rho}_{conv.} = .911$, $s_{conv.} = .955$), accounting for their similar performance.

Ultimately, Fig. 13 supports our assertion that constraining the feature subspace dimensionality to $d = 2$ does not significantly degrade performance for this application. Utilizing this strategy reduces the number of possible feature subspaces from 255 to 28, yielding considerable gains in efficiency during parameter optimization. Table 1 shows the training times for each stage of the two algorithms. By constraining the feature subspaces, the total training time can be reduced by over 82%.

	Parameter Optimization Time (hours)	Ensemble Training Time (hours)	Total Time (hours)	Total Relative Reduction
RS-SVC	303.84	5.75	309.59	N/A
CS-RS-SVC	360.00	4.25	364.25	N/A
RS-SVC d=2	41.90	7.25	49.15	84.1%
CS-RS-SVC d=2	59.29	6.40	65.69	82.0%

Table 1 – Runtimes for the optimization and training of RS-SVC and CS-RS-SVC ensembles, along with their relative reduction when constrained dimensionality ($d = 2$) is used. All systems were trained on a dual-core Intel 6600 2.4 GHz processor w/ 4GB RAM.

VI. Conclusion

We have compared the performance of conventional and cost-sensitive Support Vector Classifiers (SVCs) in singleton as well as ensemble detection systems applied to a real-world detection application that requires nonzero detection probabilities at ultra-low false alarm rates. Empirical evidence has shown that while bagging enhances the performance of singleton (conventional) SVC systems, the Random Subspace method provides significantly enhanced gains in detection rate over low FAR regions. The effectiveness of this method can be attributed largely to the base classifier diversity resulting from random sampling of both training data as well as feature subspaces.

The CS-SVC incorporates an additional parameter that enables optimization of detection performance at ultra-low false alarm rates. While bagging of CS-SVCs did not perform significantly better than bagging of conventional SVCs, the Random Subspace method did yield a significant gain in performance over low FAR intervals when its base classifiers were made cost-sensitive. This methodology appeared to more effectively leverage the additional flexibility of CS-SVCs than bagging alone.

The cost-sensitive CS-RS-SVC outperformed the conventional RS-SVC in terms of both detection rate and expected cost across ultra-low false alarm regions. Of even greater significance was its increase in detection rate at a FAR of zero by a factor of more than four.

Though the parameter optimization procedures are quite expensive for the CS-RS-SVC, we found that restricting the pool of feature subspaces to dimension $d = 2$ reduces the number of SVC parameter optimizations by 89% and overall training time by 82% without sacrificing performance. This result was further supported by an examination of base classifier strength and correlation over the negative class. It is important to note, however, that the optimal value of d is likely data dependent, and may rely at least partially upon the presence or absence of correlation among the various feature combinations.

Acknowledgment

The authors are grateful to David Knapp, John Buyer, Larry Hiller, and Marshall Mugge for providing the Hidden Signal dataset and features.

References

[1] H. G. Chew, R. E. Bogner, and C. C. Lim, "Dual-ν support vector machine with error rate and training size biasing," in Proc. International Conference on Acoustics, Speech, and Signal Processing, pp. 1269–1272, 2001.

[2] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Controlling false alarms with support vector machines," in Proc. International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, 2006.

[3] B. Y. Chen, T. L. Hickling, M. Krnjajić, W. G. Hanley, G. Clark, J. Nitao, D. Knapp, L. Hiller, and M. Mugge "Multi-Layer Perceptrons and Support Vector Machines for Detection Problems with Low False Alarm Requirements: an Eight-Month Progress Report," LLNL, UCRL-TR-227939, February 2007.

[4] C. Elkan, "The Foundations of Cost-Sensitive Learning," in Proc. of the Seventeenth International Joint Conference on Artificial Intelligence, pp. 973-978, 2001.

[5] L. Breiman, "Bagging predictors," Machine Learning, Vol. 24, No. 2, 1996.

[6] G. Valentini, T. G. Dietterich, "Bias-variance analysis of Support Vector Machines for the development of SVM-based ensemble methods," Journal of Machine Learning Research, 5, pp. 725-775, 2004.

[7] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, pp. 832-844, August 1998.

[8] L. Breiman, "Random Forests," Machine Learning, Vol. 45, No. 1, pp. 5-32, 2001.

[9] P. Domingos, "MetaCost: a general method for making classifiers cost-sensitive," in Proc. of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 155-164, August 15-18, 1999.

[10] W. Fan, S. J. Stolfo, J. Zhang, P. K. Chan, "AdaCost: Misclassification Cost-Sensitive Boosting," in Proc. of the Sixteenth International Conference on Machine Learning, pp. 97-105, June 27-30, 1999.

[11] H. Masnadi-Shirazi, N. Vasconcelos, "Asymmetric Boosting," in Proc. of the Twenty-Fourth International Conference on Machine Learning, pp. 609-619, 2007.

[12] B. E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classiers," In Proc. of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152, 1992.

[13] C. Cortes, V. Vapnik, "Support-vector network," Machine Learning, Vol. 20, pp. 273-297, 1995.

[14] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," Advances in Neural Information Processing Systems 5, pp. 147-155, 1993.

[15] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, 27, pp. 861-874, 2006.

[16] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in Proc. of the Fourteenth International Joint Conference on Artificial Intelligence, 2, 12, pp. 1137-1143, 1995.

[17] J. A. Nelder, R. Mead, "A simplex method for function minimization," Computer Journal, Vol. 7, pp. 308-313, 1965.

[18] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001, Software: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[19] C. Drummond and R. Holte, "Explicitly Representing Expected Cost: An Alternative to ROC Representation", in Proc. of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.