TRAINING A SUPPORT VECTOR MACHINE TO CLASSIFY SIGNALS IN A REAL ENVIRONMENT GIVEN CLEAN TRAINING DATA

Kevin Jamieson, Maya R. Gupta, Eric Swanson, and Hyrum S. Anderson

Dept. of Electrical Engineering, University of Washington, Seattle WA 981095

ABSTRACT

When building a classifier from clean training data for a particular test environment, knowledge about the environmental noise and channel should be taken into account. We propose training a support vector machine (SVM) classifier using a modified kernel that is the expected kernel with respect to a probability distribution over channels and noise that might affect the test signal. We compare the proposed expected SVM to an SVM that ignores the environment, to an SVM that trains with multiple random samples of the environment, and to a quadratic discriminant analysis classifier that takes advantage of environment statistics (Joint QDA). Simulations classifying narrowband signals in a noisy acoustic reverberation environment indicate that the expected SVM can improve performance over a range of noise levels.

***Index Terms*—** support vector machine, classification, quadratic discriminant analysis, sonar, speech

1. INTRODUCTION

A common problem in building classifiers for signals is that the environment the classifier will be used in is not known when the training samples are being collected, and thus it can be difficult to collect training samples subjected to the same channel and additive noise that the test signals will encounter. For example, speech recognition systems may be used in an office, car or street environment, each with different reverberation and ambient noise effects on the test signals input to the classifier. Sonar signals are subject to a channel and noise that depends on the ocean geometry and weather. To make it possible to build classifiers for many environments, one can collect or create training signals that are as clean as possible, for example by using high-quality recording equipment for speech, or by modeling acoustic signatures for sonar. Then the problem becomes, "Given clean training signals, what is the best way to train a classifier for a given test environment?"

Here, we assume that the test environment is unknown but can be characterized as a linear time-invariant system with additive noise, where the channel h and noise w are random, and the classifier receives a test signal

$$z[k] = h[k] * x[k] + w[k], \tag{1}$$

where the channel h is assumed to be drawn from a distribution over channels with a known mean and covariance, the true test signal x is unknown but we assume its distribution depends on its true class label y and that it is drawn iid with n training example signals $\{x_i\}_{i=1}^n$, and we assume zero-mean additive noise w with power $E[ww^T] = \sigma^2 I$.

Performance can be very poor if one classifies a noisy test signal $z[k]$ with a classifier optimized for clean training pairs $\{x_i, y_i\}_{i=1}^n$, where y_i is the class label of the ith training signal. One solution, common in speech processing and recently investigated for sonar [1], is to simulate example channels and noise for the given test environment and train a classifier with simulated corrupted training signals $\{x_i * h_v + w_t\}$ generated from training signals $\{x_i\}_{i=1}^n$, channels $\{h_v\}_{v=1}^V$, and noise signals $\{w_t\}_{t=1}^T$. Thus, to capture the environment variability may require a large number $n \times V \times T$ of simulated corrupted training signals; this may be problematic for the support vector machine, whose training cost may be as severe as $O(n^3)$.

Recently, Anderson and Gupta proposed taking the channel and noise statistics into account when building a classifier from clean training data [2]. They showed how to derive a quadratic discriminant analysis classifier termed *Joint QDA* that incorporates the mean and covariance of the unknown channel and noise, and showed improved performance for two sonar signal classification problems.

In this paper, we propose a method to incorporate information about the channel and noise statistics into the training of a support vector machine (SVM) classifier given clean training signals. We derive closed-form solutions for the SVM using the popular linear kernel and radial basis function (RBF) kernel for classifying the original signals (which can be easily generalized to classifying any linear transform of the original signals such as wavelet or Fourier coefficients), and for classifying based on Fourier subband energies, which is a common feature in sonar applications.

Experiments compare the accuracy and runtime of the derived *Expected SVM* to Joint QDA and to the more standard approach of training an SVM with simulated corrupted training signals.

This research is supported by an ONR YIP Award.

2. EXPECTED SVM

An SVM classifies a sample x based on the sign of a discriminant function $f(x) = \alpha_0 + \sum_i \alpha_i y_i K(x, x_i)$, where the ith training sample class label $y_i \in \{-1, 1\}$, $K(\cdot, \cdot)$ is the kernel (an inner product function in the implicit reproducing kernel Hilbert space), and α_0 and the α_i's are scalars that are learned by the SVM to minimize the (regularized) training error for the training pairs $\{x_i, y_i\}_{i=1}^n$.

A kernel K measures the similarity of its two arguments. Two standard kernels are the linear kernel:

$$K(x, x_i) = x^T x_i, \tag{2}$$

and RBF kernel with bandwidth γ, which can equivalently be expressed as a Gaussian function evaluated at x with mean x_i and spherical covariance $\gamma^2 I$:

$$K(x, x_i) = \mathcal{N}(x; x_i, \gamma^2 I). \tag{3}$$

The key insight of this paper is that one can view the problem of classifying corrupted test samples as a question of how to best define the similarity $K(z, x_i)$ between a corrupted test sample z and a clean training sample x_i. We propose answering this question by defining a random corrupted training signal z_i for each deterministic training signal x such that

$$z_i = x_i * h_i + w_i, \tag{4}$$

where we assume that h_i and w_i are a random channel and random noise drawn from the test environment's distribution of random channels and noise. Then we propose to test the SVM classifier using the *expected corrupted test kernel*:

$$K(z, x_i) \stackrel{\triangle}{=} E_{z_i}\left[\tilde{K}(z, z_i)\right], \tag{5}$$

where $\tilde{K}$ is any kernel function, such as the linear kernel (2) or RBF kernel (3).

To take the test environment into account when training the SVM with the clean data, we propose training with the *expected corrupted training kernel*:

$$K(x_i, x_j) \stackrel{\triangle}{=} E_{z_i, z_j}\left[\tilde{K}(z_i, z_j)\right], \tag{6}$$

where z_i, z_j are given by (4).

The expected corrupted training kernel can be interpreted as simulating all possible corruptions of x_i and x_j and considering the average similarity between all possible corrupted versions. The expected test kernel (5) can be expressed as (6) by treating the deterministic z as random with a Dirac distribution on z. Because the set of kernels is convex, the expected kernel is also a kernel.

In the next sections we derive the expected kernels for four common cases: classifying signals directly or classifying subband energy features derived from the signals, and using either the expected linear kernels or the expected RBF kernels. Throughout we assume all random channels are iid and all noise is iid.

3. CLASSIFYING SIGNALS DIRECTLY

Given signals, we derive the expected test and training kernels for the linear kernel and the RBF kernel.

3.1. Expected Linear Kernels for Time Signals

The expected test kernel (5) for linear kernel (2) is:

$$E_{h_i, w_i}[z^T(x_i * h_i + w_i)] = z^T(x_i * \mu_h),$$

where μ_h is the expected channel $E_h[h]$.

The expected training kernel (6) is:

$$\begin{aligned} &E_{h_i, h_j, w_i, w_j}[(x_i * h_i + w_i)^T(x_j * h_j + w_j)] \\ &= (x_i * \mu_h)^T(x_j * \mu_h). \end{aligned}$$

These expected kernels only use information about the expected impulse response μ_h of the channel.

3.2. Expected RBF Kernels for Time Signals

To calculate the expected kernels for the RBF kernel (3) we model the additive noise as Gaussian: $w_i \sim \mathcal{N}(0, \sigma^2 I)$, and the random channel as Gausssian: $h_i \sim \mathcal{N}(\mu_h, \Sigma_h)$. Let X_i be the convolution matrix such that $\mathrm{X}_i h_i = x_i * h_i$. It follows that $p(z_i|x_i) = \mathcal{N}(z_i; \mathrm{X}_i\mu_h, \mathrm{X}_i\Sigma_h\mathrm{X}_i^T + \sigma^2 I)$.

Then the expected test kernel (5) for the RBF kernel (3) evaluated at z is

$$\begin{aligned} E_{z_i}\left[\mathcal{N}(z_i; z, \gamma^2 I)\right] &= \int_{\tilde{z}_i} \mathcal{N}(\tilde{z}_i; z, \sigma^2 I) p(\tilde{z}_i|x_i) d\tilde{z}_i \\ &= \mathcal{N}\left(z; \mathrm{X}_i\mu_h, \mathrm{X}_i\Sigma_h\mathrm{X}_i^T + \gamma^2 I + \sigma^2 I\right), \end{aligned}$$

by the product of Gaussians rule [3].

The expected training kernel (6) for the RBF kernel (3) is:

$$\begin{aligned} &\int_{\tilde{z}_i}\int_{\tilde{z}_j} \mathcal{N}(\tilde{z}_i; \tilde{z}_j, \sigma^2 I) p(\tilde{z}_i, \tilde{z}_j|x_i, x_j) d\tilde{z}_i d\tilde{z}_j \\ &= \int_{\tilde{z}_i}\int_{\tilde{z}_j} \mathcal{N}(\tilde{z}_i; \tilde{z}_j, \sigma^2 I) p(\tilde{z}_i|x_i) p(\tilde{z}_j|x_j) d\tilde{z}_i d\tilde{z}_j \\ &= \mathcal{N}(\mathrm{X}_i\mu_h; \mathrm{X}_j\mu_h, \mathrm{X}_i\Sigma_h\mathrm{X}_i^T + \mathrm{X}_j\Sigma_h\mathrm{X}_j^T + \gamma^2 I + 2\sigma^2 I), \end{aligned}$$

by successive application of the product of Gaussians rule, where we have assumed that $p(z_i, z_j|x_i, x_j) = p(z_i|x_i)p(z_j|x_j)$.

These RBF expected kernels take into account more information about the noise environment than the expected linear kernels.

4. CLASSIFYING SUBBAND ENERGY FEATURES

In this section, we consider the case that the linear or RBF kernel acts on a feature vector U_{x_i} whose components are the energies in specified Fourier subbands of x_i. The expected

energy of the noise signal in any subband is taken to be the noise power σ^2 times the length L of the test signal z.

Because $z_i = h_i * x_i + w_i$, we can characterize the mean and covariance of the random vector U_{z_i} in terms of the statistics of the random channel and noise:

$$\mu_{U_{z_i}} = U_{x_i} \cdot \mu_{U_h} + L\sigma^2 \mathbf{1} \tag{7}$$

$$\Sigma_{U_{z_i}} = (L\sigma^2)^2 I + \Sigma_{U_h} \cdot U_{x_i} U_{x_i}^T + 2L\sigma^2 \text{diag}\left(U_{x_i} \cdot \mu_{U_h}\right), \tag{8}$$

where $\cdot$ denotes the component-wise product of its left and right arguments, and $\mathbf{1}$ is a vector of ones.

4.1. Subband Energy Features: Expected Linear Kernels

The expected test kernel $K(U_z, U_{x_i})$ for linear kernel (2) is:

$$E_{h_i,w_i}[U_z^T U_{z_i}] = U_z^T E_{h_i,w_i}[U_{z_i}] = U_z^T \mu_{U_{z_i}}.$$

The corresponding expected linear training kernel $K(U_{x_i}, U_{x_j})$ is:

$$E_{h_i,w_i,h_j,w_j}[U_{z_i}^T U_{z_j}] = \mu_{U_{z_i}}^T \mu_{U_{z_j}}.$$

As seen in (7), these expected linear kernels only take into account the expected subband energies of the channel and the expected noise energy.

4.2. Subband Energy Features: Expected RBF Kernels

To calculate the expected RBF kernels between subband energy feature vectors, we make the maximum entropy assumption that the random corrupted subband energy feature vector U_{z_i} is drawn from a Gaussian distribution with the mean and covariance stated in (7) and (8). Then the expected RBF test kernel $K(U_z, U_{x_i})$ is:

$$\begin{aligned}
&\int_{\tilde{U}_{z_i}} \mathcal{N}(\tilde{U}_{z_i}; U_z, \gamma^2 I) p(\tilde{U}_{z_i}|x_i) d\tilde{U}_{z_i} \\
&= \int_{\tilde{U}_{z_i}} \mathcal{N}(\tilde{U}_{z_i}; U_z, \gamma^2 I) \mathcal{N}(\tilde{U}_{z_i}; \mu_{U_{z_i}}, \Sigma_{U_{z_i}}) d\tilde{U}_{z_i} \\
&= \mathcal{N}(U_z; \mu_{U_{z_i}}, \Sigma_{U_{z_i}} + \gamma^2 I),
\end{aligned} \tag{9}$$

where the last line follows by the product of Gaussians rule, and $\mu_{U_{z_i}}$, $\Sigma_{U_{z_i}}$ are given in (7) and (8).

The expected RBF training kernel $K(U_{x_i}, U_{x_j})$ is:

$$\begin{aligned}
&\int_{\tilde{U}_{z_i}} \int_{\tilde{U}_{z_j}} \mathcal{N}(\tilde{U}_{z_i}; \tilde{U}_{z_j}, \gamma^2 I) p(\tilde{U}_{z_i}; \tilde{U}_{z_j}|U_{x_i}, U_{x_j}) d\tilde{U}_{z_i} d\tilde{U}_{z_j} \\
&= \int_{\tilde{U}_{z_i}} \int_{\tilde{U}_{z_j}} \mathcal{N}(\tilde{U}_{z_i}; \tilde{U}_{z_j}, \gamma^2 I) \mathcal{N}(\tilde{U}_{z_i}; \mu_{U_{z_i}}, \Sigma_{U_{z_i}}) \\
&\qquad \mathcal{N}(\tilde{U}_{z_j}; \mu_{U_{z_j}}, \Sigma_{U_{z_j}}) d\tilde{U}_{z_i} d\tilde{U}_{z_j} \\
&= \mathcal{N}(\mu_{U_{z_i}}; \mu_{U_{z_j}}, \Sigma_{U_{z_i}} + \Sigma_{U_{z_j}} + \gamma^2 I)
\end{aligned} \tag{10}$$

by successive application of the product of Gaussians rule.

These expected RBF kernels take into account the covariance of the channel subband energies, as well as the mean channel subband energies and the noise power.

5. EXPERIMENTS AND RESULTS

Experiments were run with the proposed expected SVM classifier using the RBF kernel on subband features as given in (9) and (10). The simulation is the same as in Anderson et al. [2] except we use 20 training signals rather than 1000 training signals, and subband energy features are used in place of power features. Narrowband signals from two classes were simulated by randomly perturbing the placement of poles in the z-transform domain; three choices of pole separation to three classification problems ranging from easy to hard; energies for two frequencies were used as subband energy features. At test the signals were propagated through a shallow water channel before being classified, where the channels h were drawn iid using the *CASS Eigenray* routine in the Sonar Simulation Toolset for the bathymetry.

The expected SVM classifier was compared to joint QDA [2] and to three RBF SVMs: blind, informed [2], and *simulated corrupted blind* (SCB) [1]. The blind RBF SVM trains on the clean training data and treats the corrupted test sample as though it was clean. The informed RBF SVM trains on the clean training data, but before testing subtracts off the expected noise energy and divides by the channel energy. SCB SVM trains on simulated corrupted training samples that are created by selecting N random channels (different draw of channels for each run of the simulation), then corrupting each of the clean training samples with each of the N channels and a random draw of the noise. Thus the SCB SVM trains on N times as many training samples as any of the other considered classifiers. Preliminary experiments with $N = 5$ to $N = 12$ did not show a difference in performance. For the results reported here we used $N = 10$, which takes 104 minutes for a complete run of the simulation on a 2.4GHz machine. (The SCB method is similar to the method of *virtual examples* in [4] for invariant SVMs, however, the simulated corrupted examples are generated from a stochastic model is rather than deterministically.)

Each run of the simulation was performed on 2000 randomly drawn signals. For each run, $n = 20$ training samples were randomly selected, and the other 1980 signals were treated as test samples. Each test sample was convolved with an iid random draw from the 2000 simulated channel impulse responses, and subjected to iid Gaussian noise over a range of SNR's.

The bandwidth γ parameter in (3) was independently selected for each RBF SVM classifier and for each SNR using leave-one-out cross validation (LOOCV) on the 20 training samples. For SCB SVM, the cross validation is performed for each of the 10 random channels, leaving each cv split with

180 training signals and 20 test signals. To make the comparison similar time-wise to SCB SVM, for the LOOCV the expected SVM takes left-out training sample in the LOOCV and corrupts it 100 times with a set of 100 randomly drawn channels, so that the γ is chosen to minimize error on the environment. This LOOCV makes the expected SVM take 105 minutes on average for a run of the simulation (mostly due to time to perform convolutions), compared with SCB SVM taking 104 minutes to complete a run.

Results averaged over 75 runs of the experiment are shown in Figure 1. The proposed expected SVM is consistently the best for SNRs greater then 0 dB. At lower SNR all classifiers performed poorly, although joint QDA and the informed SVM appear to be the most robust.

6. DISCUSSION

We presented an *expected kernel* method to take into account the test environment when using and training an SVM from clean training samples. We showed that for an RBF kernel the expected kernel will take into account the expected channel, channel covariance, and noise power. Experimental results showed that the expected SVM can work better than simulating corrupted training samples for the same total training time, and our experiments lead us to hypothesize that the expected SVM can work better with much shorter training time for many practical situations.

Here we treated z_i and z_j as encountering independent random corruptions, which is analogous to the common practice of simulating example channels independently. However, we hypothesize that better results are possible if one treats the z_i and z_j as sharing the same unknown random channel h and noise w, but we leave this for future work.

7. REFERENCES

[1] A. J. Llorens, T. L. Philip, I. W. Schurman, and C. R. Lorenz, "Enhancing passive automation performance using an acoustic propagation simulation," *J. Acoust. Soc. Am*, vol. 125, no. 4, pp. 2577, April 2009.

[2] H. S. Anderson and M. R. Gupta, "Joint deconvolution and classification with applications to passive acoustic underwater multipath," *J. Acous. Soc. Am.*, vol. 124, no. 5, pp. 2973–2983, November 2008.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[4] D. Decoste and B. Schölkopf, "Training invariant support machines," *Mach. Learn.*, vol. 46, pp. 161–190, 2002.

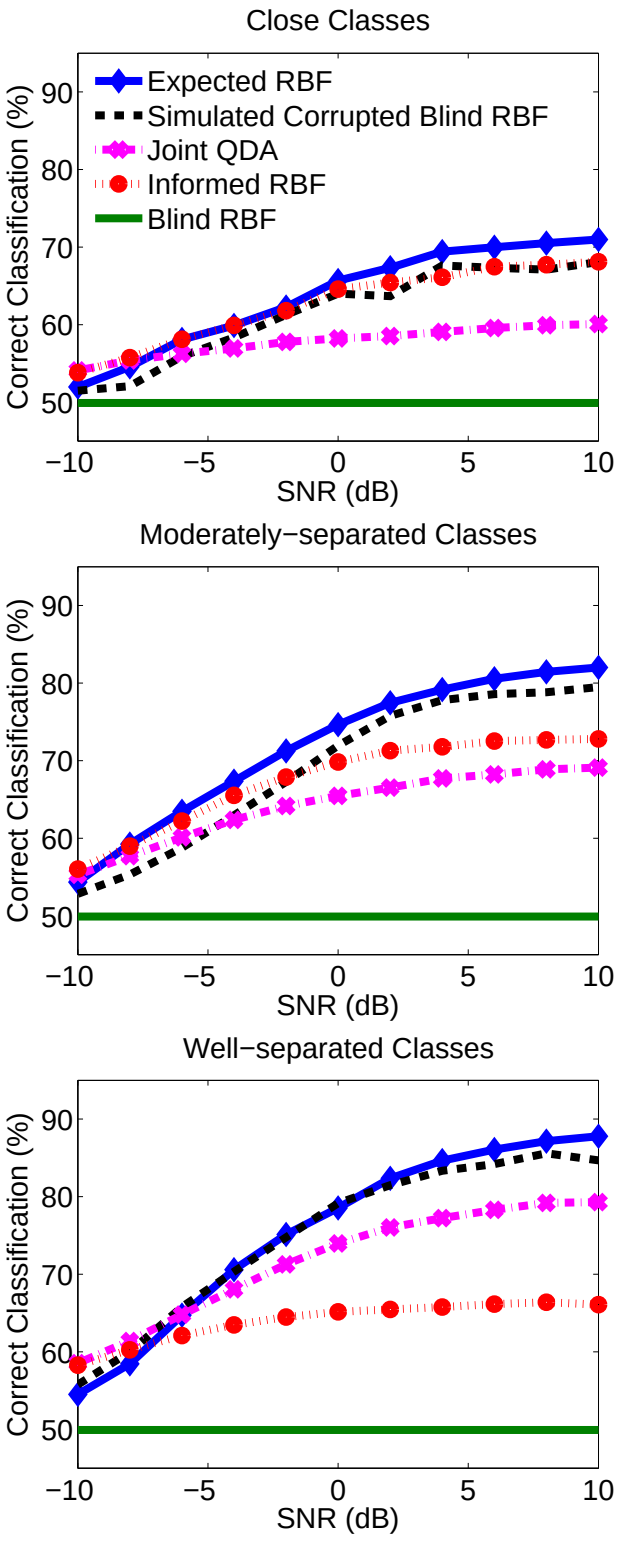

Fig. 1. Results of classifying subband energy features where the classes range from difficult to separate (top) to easy to separate (bottom).