BAYESIAN TRANSFER LEARNING FOR NOISY CHANNELS

*Nathan Parrish, Maya R. Gupta**

Department of Electrical Engineering
University of Washington
Seattle, WA

ABSTRACT

We consider the problem of classifying a signal that is the output of a linear, time-invariant channel in the presence of additive noise, given two distinct sets of labeled data: one dataset of examples of the signals input to the channel, and a second dataset of example signals corrupted by the channel. We propose a distribution-based Bayesian quadratic discriminant analysis classifier that uses the input examples along with a model for the channel to form a prior for the likelihood of the output examples. Preliminary experiments with this proposed transfer BDA classifier show that it effectively uses both sets of data and is also robust to errors in channel modeling.

***Index Terms*—** Bayesian methods, classification algorithms, machine learning algorithms, signal processing algorithms, multipath channels

1. INTRODUCTION

A recent topic of interest in classification is the transfer of knowledge gained in one domain to a classification task in a new domain [1]. In many signal processing applications, such a scenario arises because a signal to be classified, z, is the output of a random unknown linear, time-invariant (LTI) channel, h, with additive noise, w:

$$z = h * x + w. \tag{1}$$

For example, the signal x could be an acoustic or seismic signal which propagates through some unknown multipath channel. Alternatively, the signal to be classified could be an image and the channel, h, could represent some unknown point spread function. In the remainder of this paper, we will refer generally to test signals z as residing in the *target* domain and signals x as residing in the *source* domain.

We note that within the field of *transfer learning* there are many approaches to make use of both source and target domain training samples [1], however, our approach differs from previous literature in that we assume we also have a model linking the source and target domain, as given by (1). Other related research has focused on a special case of our problem formulation, where the learner is given a set of source domain training examples, $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{n_s}$ where $x_i \in R^{d_s}$ and $y_i \in \mathcal{Y}$ are the i^{th} feature vector and class label, that it must use to classify the test point z. Researchers have proposed several approaches to this problem. One such approach is to first perform blind deconvolution of z followed by classification, and there are many examples of this approach in the literature, for example [2].

Other approaches to classifying z make use of side information in the form of estimated or simulated channel examples, $\{\hat{h}_i\}$, that are assumed to be drawn iid from the same distribution as the random channel in (1). The method of 'virtual training examples' creates a virtual training set with feature vectors $\hat{z}_i = x_i * \hat{h}_i$ using the source examples in $\mathcal{X}$ and the channel examples, and then uses these examples to train a target domain classifier using standard classification methods [3, 4]. Alternatively, the channel can be modeled as a realization of a random Gaussian variable, and the $\hat{h}_i$ can be used to estimate the mean and covariance of the Gaussian distribution. In [5], the training examples and channel model are used to train a quadratic discriminant analysis (QDA) classifier in the target domain, which is referred to as *joint QDA*, and experiments showed that joint QDA outperforms both blind deconvolution then classification and joint deconvolution and classification of z. Finally, in [6] a Gaussian random channel model is used to adapt the kernel function used to train an SVM classifier for classification of z using the training set $\mathcal{X}$. Results showed that in many cases this expected kernel approach is comparable in performance to a local form of joint QDA.

In this paper we extend the above formulations to consider the more general case where, in addition to the training set $\mathcal{X}$, we also have a labeled target domain training set $\mathcal{Z} = \{(z_j, y_j)\}_{j=1}^{n_t}$ $z_j \in R^{d_t}$, $y_j \in \mathcal{Y}$. We assume that the elements in $\mathcal{Z}$ are generated according to $z_j = h_j * x_j + w_j$ where the x_j are unknown, but are drawn iid from the same distribution as the members of $\mathcal{X}$ and that the h_j are realizations of the random channel drawn iid from the same distribution as h. Like several of the other approaches above, we also assume that we have a mechanism for estimating the mean, $\hat{\mu}_h$, and covariance, $\hat{\Sigma}_h$, of the channel distribution (ie., from channel examples). However, unlike the above approaches,

*Thanks to the United States Office of Naval Research for funding.

we make the more practical assumption that these estimates may be biased or inaccurate estimates of the true channel parameters, and build robustness to inaccurate channel modeling into our algorithm.

2. DISTRIBUTION-BASED BAYESIAN QDA FOR TRANSFER LEARNING

In order to make use of both the source and target domain training sets, we use the framework of distribution-based Bayesian QDA to classify test point z. Distribution-based Bayesian QDA was first proposed by Srivistava *et al.* [7] as an alternative to regularization when the estimation of model parameters in QDA is ill-posed. QDA estimates one maximum-likelihood Gaussian distribution for each class. Alternatively, distribution-based Bayesian QDA computes the expected Gaussian distribution for each class.

Our Bayesian QDA differs significantly from the work of Srivistava in our random model for the class-conditional Gaussians. Specifically, we use the source domain training data $\mathcal{X}$ to form a prior probability over the space of Gaussian distributions for each class. In this way, the source domain data regularizes the estimates of the target domain class conditional distributions. In addition, by using the source domain data to form a prior, the prior's hyperparameters can be used to create robustness to inaccurate channel modeling.

Distribution-based Bayesian QDA models the class-conditional signal likelihood $p(z|Y=y)$ as a random Gaussian distribution, $p(z|Y=y) = N_{Z|y}(z)$ with realization $\mathcal{N}_{Z|y}(z)$, and classifies the test point z according to the class which maximizes the expected a-posteriori probability:

$$\hat{y} = \arg\max_{y\in\mathcal{Y}} \mathrm{E}_{N_{Z|y}}\left[N_{Z|y}(z)\right]\hat{p}(Y=y), \tag{2}$$

where the last term in (2) is an estimate of the a-priori probability of class y.

The expected distribution in (2) can be written as

$$\mathrm{E}_{N_{Z|y}}\left[N_{Z|y}(z)\right] = \int_M \mathcal{N}_{Z|y}(z) r(\mathcal{N}_{Z|y}) dM, \tag{3}$$

where M is an appropriate measurable space and $r(\mathcal{N}_{Z|y})$ is the probability of the Gaussian distribution. As was done in [7], we set M to be the set of all Gaussian distributions parameterized by $\mu_{z|y} \in R^d$ and $\Sigma_{z|y} \subset S$ where S is the cone of $d \times d$ positive semi-definite matrices. Furthermore, we set the differential element, $dM = d\mu_{z|y} d\Sigma_{z|y}$.

We propose to let $r(\mathcal{N}_{Z|y})$ in (3) be the posterior probability density of $\mathcal{N}_{Z|y}$ given the training sets $\mathcal{X}$ and $\mathcal{Z}$:

$$r(\mathcal{N}_{Z|y}) = p(\mathcal{N}_{Z|y}|\mathcal{Z},\mathcal{X}) = \gamma_y^{-1} p(\mathcal{Z}|\mathcal{N}_{Z|y},\mathcal{X}) p(\mathcal{N}_{Z|y}|\mathcal{X}). \tag{4}$$

The γ_y^{-1} term in (4) is a normalization constant, the second term is the likelihood function of data in $\mathcal{Z}$, and the third term is the prior distribution of $\mathcal{N}_{Z|y}$ given by the source data. If we represent $\mathcal{Z}_y$ and $\mathcal{X}_y$ as the elements of the training sets with only class label y then we can expand (4) as:

$$r(\mathcal{N}_{Z|y}) = \gamma_y^{-1} p(\mathcal{Z}_y|\mathcal{N}_{Z|y}) p(\mathcal{N}_{Z|y}|\mathcal{X}_y) \prod_{\substack{i=1 \\ i\neq y}}^{|\mathcal{Y}|} p(\mathcal{Z}_i|\mathcal{N}_{Z|y},\mathcal{X}), \tag{5}$$

where we have assumed that $p(\mathcal{N}_{Z|y})$ is independent of the training samples in $\mathcal{X}$ that are not in class y given the samples in y. The terms within the product in (5) are likelihood functions for the target data in classes other than y given the conditional distribution $\mathcal{N}_{Z|y}$ and the source data. Although it may be possible to use these likelihoods to improve the estimation of the posterior probability, in this study we assume $p(\mathcal{Z}_i|\mathcal{N}_{Z|y},\mathcal{X}) = p(\mathcal{Z}_i)$ for $i \neq y$; therefore, (5) reduces to:

$$r(\mathcal{N}_{Z|y}) \sim \gamma_y^{-1} p(\mathcal{Z}_y|\mathcal{N}_{Z|y}) p(\mathcal{N}_{Z|y}|\mathcal{X}_y) \tag{6}$$

The first probability in (6) is a Gaussian likelihood function, and can be written in terms of the mean $\mu_{z|y}$ and covariance $\Sigma_{z|y}$ of $\mathcal{N}_{Z|y}$ as

$$\begin{aligned} p\left(\mathcal{Z}_y|\mathcal{N}_{Z|y}(\mu_{z|y},\Sigma_{z|y})\right) &= (2\pi)^{\frac{-d_t n_{sy}}{2}} |\Sigma_z|^{\frac{-n_{sy}}{2}} \\ &\cdot e^{\frac{-n_{sy}}{2}(\mu_{z|y}-\bar{z}_y)^T \Sigma_{z|y}^{-1}(\mu_{z|y}-\bar{z}_y)} e^{\frac{-1}{2}\mathrm{tr}(V_y \Sigma_{z|y}^{-1})} \end{aligned} \tag{7}$$

where $\bar{z}_y$ is the sample mean from the training set $\mathcal{Z}_y$ with n_{sy} elements and $V_y = \sum_{i=1}^{n_{sy}} (z_i - \bar{z}_y)(z_i - \bar{z}_y)^T,\ z_i \in \mathcal{Z}_y$.

To solve the posterior distribution in (6), we must model a prior, $p(\mathcal{N}_{Z|y}|\mathcal{X}_y)$. We make the assumption that the probability of the mean and covariance of $\mathcal{N}_{Z|y}$ can be decomposed as:

$$p(\mathcal{N}_{Z|y}|\mathcal{X}_y) = p(\mu_{z|y}|\Sigma_{z|y},\mathcal{X}_y) p(\Sigma_{z|y}|\mathcal{X}_y).$$

Specifically, we use a Normal-inverse Wishart distribution for the prior:

$$p(\mathcal{N}_{Z|y}|\mathcal{X}_y) = \mathcal{N}\left(\mu_{z|y}; m_y, \frac{\Sigma_{z|y}}{t_y}\right) IW(\Sigma_{z|y}; B_y, \alpha_y). \tag{8}$$

The joint distribution in (8) has four hyperparameters, and we note that the mean of the inverse Wishart distribution is $\frac{B_y}{\alpha_y - d_t - 1}$.

We set m_y and B_y in (8) to match the mean and scaled covariance of Z given the channel model (1), the side information about the channel ($\hat{\mu}_h$ and $\hat{\Sigma}_h$), our estimate of the noise variance, and estimates of the class conditional mean and covariance $\hat{\mu}_{x|y}$ and $\hat{\Sigma}_{x|y}$ from $\mathcal{X}_y$. That is, we set

$$m_y = \hat{\mu}_h * \hat{\mu}_{x|y}, \text{ and} \tag{9}$$

$$\begin{aligned} B_y &= (\alpha - d_t - 1) \\ &\cdot \left((\hat{\Sigma}_h + \hat{\mu}_h \hat{\mu}_h^T) * * (\hat{\Sigma}_{x|y} + \hat{\mu}_{x|y} \hat{\mu}_{x|y}^T) - m_y m_y^T + \hat{\sigma}_w^2 I \right), \end{aligned} \tag{10}$$

where $**$ represents 2-D convolution. We leave the discussion of setting parameters t_y and α_y to the next section.

The product of (7) and (8) gives the posterior density of the random distribution $\mathcal{N}_{Z|y}$. The posterior distribution is

$$r(\mathcal{N}_{Z|y}) = \mathcal{N}\left(\mu_{z|y}; m'_y, \frac{\Sigma_{z|y}}{t'_y}\right) \mathrm{IW}(\Sigma_{z|y}; B'_y, \alpha'_y) \quad (11)$$

with parameters

$$m'_y = \frac{n_{sy}\bar{z}_y + t_y m_y}{n_{sy} + t_y},\ t'_y = n_{sy} + t_y,$$
$$B'_y = B_y + \frac{n_{sy} t_y}{n_{sy} + t_y}(\bar{z}_y - m_y)(\bar{z}_y - m_y)^T + V_y,$$
$$\alpha'_y = \alpha_y + n_{sy}.$$

We can now evaluate the expected value of the random normal distribution $\mathrm{E}_{N_{Z|y}}[N_{Z|y}(z)]$ by substituting the posterior density (11) into (3). Defining $C_y \triangleq \frac{t'_y}{2(t'_y+1)}(m'_y - z)(m'_y - z)^T + \frac{1}{2}B'$, the integral in (3) evaluates to

$$\mathrm{E}_{N_{Z|y}}[N_{Z|y}(z)] = \frac{|B'_y|^{\alpha'_y/2}\Gamma_{d_t}\left(\frac{\alpha'_y+1}{2}\right) t'^{d_t/2}_y}{(2\pi)^{d_t/2}2^{\alpha'_y d_t/2}\Gamma_{d_t}(\alpha'_y/2)(t'_y+1)^{d_t/2}|C_y|^{(\alpha'_y+1)/2}}. \quad (12)$$

Substituting (12) into (2) gives the final classification rule, and we call the resulting classifier *transfer BDA*.

3. CHANNEL MODEL ROBUSTNESS VIA CROSS VALIDATION

From (8), we can see that the parameters t_y and α_y control the variance of the priors for the mean and covariance, respectively. Since the means for the prior values are derived from the source data and the channel side information, as shown in (9) and (10), these parameters represent our level of confidence in the source data and channel information.

We can allow the data to select these parameters for us by performing cross validation using the target data training set, $\mathcal{Z}$, and then choosing the t_y and α_y which produce the lowest error. As opposed to cross validating t_y and α_y individually for each class, we choose one set of K cross validation parameters $\{\nu_k\},\ k = 1, ...K$ and at the k^{th} cross validation trial set $t_y = \nu_k \frac{n_{sy}}{n_{ty}}$ and $\alpha_y = t_y + d_t + 1$ where n_{sy} and n_{ty} are the numbers of source and target domain training examples for class y. In the case that two or more cross validation parameters tie, we choose the largest, as ties are most likely to occur when there are few target training examples to cross validate over, and thus we should place more reliance on the source training data.

4. SIMULATION EXPERIMENT

We test the performance of transfer BDA by performing a binary classification simulation similar to that in [5]. A source training set of 150 signals, $\{x_i\}$, is generated iid Gaussian with class conditional parameters given in Table 1. Target training data and test data are generated for each class by first drawing an iid sample from the class conditional source distribution, then convolving the source signal with an iid sample from the channel distribution, and AWGN is added (with the variance fixed) to achieve the desired signal to noise ratio (SNR) computed as $\mathrm{SNR} = 10\ \log_{10}(E[z^2]/\sigma_w^2)$. In all cases, the a-priori probability for each class is 0.5.

Table 1. Class conditional mean and covariance.

$\mu_{x\|y}[m]$	$\Sigma_{x\|y}[m,p]$
Class 1	
$\frac{1}{4}\ \mathrm{square}(6\pi m/100)$	$\frac{1}{100}\left(\delta[m-p] + e^{-\lvert m-p\rvert/20}\right)$
Class 2	
$\frac{1}{4}\ \sin(6\pi m/100)$	$\frac{1}{100}\left(\delta[m-p] + e^{-(m-p)^2/10}\right)$

We model a random Laplacian channel with mean μ_h and scale parameter b_h; that is, the distribution of h is

$$p(h[m]|\mu_h[m], b_h[m]) = \frac{1}{2b_h[m]} e^{\frac{|-h[m]-\mu_h[m]|}{b_h[m]}}. \quad (13)$$

We set $\mu_h[m] = \delta[m] - 0.6\delta[m-49] + 0.1\delta[m-99]$, $b_h[m] = 0.2e^{-0.024m}$, and $m = 0, ..., 99$.

Our algorithm, and several of those we compare to, also requires a mechanism from which to estimate the channel mean and covariance. In order to do this, we generate $i = 1, ..., 20$ noisy channel examples according to

$$\hat{h}_i = h_i + \epsilon_i, \quad (14)$$

where each h_i is an iid realization of (13) and ϵ_i is a zero-mean AWGN noise signal with covariance $\sigma_\epsilon^2 I$. The addition of the noise, ϵ_i, to the channel examples models the likely scenario that the given channel examples are noisy examples of the real channel taken from field measurements or generated by a modeling tool such as *the Sonar Simulation Toolset* [8]. Therefore, $\sigma_\epsilon^2 = 0$ is the optimistic scenario that the channel measurements are noise free. We use these twenty channel examples to estimate the mean of h and the diaganol of the covariance of h.

For comparison, we also plot the performance of several other classifiers. We compared to two methods that use only the target training data $\mathcal{T}$: regularized discriminant analysis (RDA) [9] and a Gaussian radial basis function SVM. We also compare to joint QDA, which uses the source training data and the channel examples (14) to build a QDA classifier in the target domain [5]. Finally, we plot the performance of an SVM using the method of virtual examples. This method was implemented using a pooled training data set consisting of both the target training data $\mathcal{Z}$ as well as 150 labeled virtual examples $\hat{z}_i = x_i * \hat{h}_i$ generated by convolving the source training data with channel examples (14). Each source signal

was used to create one virtual example by randomly selecting one of the twenty channel examples. One could generate multiple virtual examples from each source signal; however, this could degrade classification performance as the additional virtual examples will mute the influence of the target examples. This virtual example method is the closest comparison to transfer BDA in that it uses both the target and source training data. For both SVM methods, we cross-validate over the target data to select the kernel bandwidth.

Results for all methods with two different values of channel estimation error, and at two different SNR levels are shown in Tables 2 and 3. These tables give the average test error over twelve experimental runs with different source and target training sets as well as different channel examples. Results highlighted in boldface tie for the lowest mean error according to a signed Wilcoxon rank test with a 5% significance level.

Results show that the scarcity of target data results in poor performance of the methods which rely solely on target examples. In general, transfer BDA outperforms all other methods, particularly in Table 3, when the channel estimates are noisier.

Table 2. Error rate for the classification of $z = h*x+w$ when the channel examples (14) are generated with $\sigma_\epsilon^2 = 0.02$.

Number of Target Samples	Transfer BDA	RDA with target data	SVM with target data	Joint QDA with source data	SVM with pooled target data and virtual examples
SNR = 0 dB					
10	**12.4**	44.2	43.4	**12.7**	21.2
20	**14.4**	42.2	47.6	**12.2**	**18.1**
40	**13.0**	35.7	45.4	**13.0**	16.8
SNR = 6 dB					
10	**5.4**	36.3	44.4	6.7	15.4
20	**5.8**	31.3	42.6	7.4	13.6
40	**5.4**	30.0	39.4	6.9	11.6

5. CONCLUSION

We have presented an approach to classifying the noisy output of a random channel. Transfer BDA is able to make use of data from both the source domain and target domain, as well as from channel examples, in order to improve performance over approaches that rely on target data alone. Most importantly, by using the source data to model a prior with flexible hyperparameters, our approach exhibits a degree of robustness to channel estimation and modeling errors. Results show that transfer BDA outperforms other channel robust approaches, particularly in the presence of channel estimation errors.

Table 3. Error rate for the classification of $z = h*x+w$ when the channel examples (14) are generated with $\sigma_\epsilon^2 = 0.05$.

Number of Target Samples	Transfer BDA	RDA with target data	SVM with target data	Joint QDA with source data	SVM with pooled target data and virtual examples
SNR = 0 dB					
10	**18.7**	45.1	47.2	24.0	26.5
20	**17.2**	39.6	43.4	22.7	**19.6**
40	**17.8**	34.2	42.0	23.3	**20.9**
SNR = 6 dB					
10	**9.6**	41.7	47.3	18.1	32.5
20	**10.4**	23.7	45.6	17.0	24.1
40	**8.0**	32.2	42.0	17.4	20.6

6. REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge Data Engineering*, vol. 22, no. 10, pp. 1345 –1359, oct. 2010.

[2] D. Kundur, D. Hatzinakos, and H. Leung, "Robust classification of blurred imagery," *IEEE Trans. Image Proc.*, vol. 9, no. 2, pp. 243 –255, Feb. 2000.

[3] N. Dasgupta and L. Carin, "Time-reversal imaging for classification of submerged elastic targets via gibbs sampling and the relevance vector machine," *J. Acous. Soc. Am.*, vol. 117, no. 4, pp. 1999–2011, 2005.

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

[5] 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, 2008.

[6] H. S. Anderson, M. R. Gupta, E. R. Swanson, and K Jamieson, "Channel-robust classifiers," *To appear in IEEE Trans. Signal Proc.*

[7] S. Srivastava, M. R. Gupta, and B. A. Frigyik, "Bayesian quadratic discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1287–1314, 2007.

[8] R. P. Goddard, "The Sonar Simulation Toolset, Release 4.6: Science, Mathematics, and Algorithms," Tech. Rep. A352884, University of Washington Applied Physics Lab, 2008.

[9] J. H. Friedman, "Regularized discriminant analysis," *J. Am. Stat. As.*, vol. 84, no. 405, pp. 165–175, 1989.