Supplementary Documentation

I. Derivation of the Gradient

We take $\phi(.)$ to be the RBF function, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma}\right)$, with σ the parameter to be optimized. And, we consider the case where each class distribution is modeled by a single Gaussian distribution. The derivations for the subclass case follows immediately from the ones given below.

The gradient of our criterion $Q(.)$, when considering the RBF kernel, is given by

$$\frac{\partial Q(\phi)}{\partial \sigma} = \frac{\partial (Q_1(\phi)Q_2(\phi))}{\partial \sigma} = \frac{\partial Q_1(\phi)}{\partial \sigma} Q_2(\phi) + \frac{\partial Q_2(\phi)}{\partial \sigma} Q_1(\phi).$$

The partial derivative of $Q_1(\phi)$ with respect to the RBF parameter σ is

$$\frac{\partial Q_1(\phi)}{\partial \sigma} = \frac{2}{C(C-1)} \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} \frac{\frac{\partial tr(\mathbf{\Sigma}_i^\Phi \mathbf{\Sigma}_k^\Phi)}{\partial \sigma}(tr(\mathbf{\Sigma}_i^{\Phi^2}) + tr(\mathbf{\Sigma}_k^{\Phi^2})) - tr(\mathbf{\Sigma}_i^\Phi \mathbf{\Sigma}_k^\Phi)(\frac{\partial tr(\mathbf{\Sigma}_i^{\Phi^2})}{\partial \sigma} + \frac{\partial tr(\mathbf{\Sigma}_k^{\Phi^2})}{\partial \sigma})}{(tr(\mathbf{\Sigma}_i^{\Phi^2}) + tr(\mathbf{\Sigma}_k^{\Phi^2}))^2}$$

Note that $\mathbf{\Sigma}_i^\Phi = \Phi(\mathbf{X}_i)(\mathbf{I} - \mathbf{1}_{n_i})\Phi(\mathbf{X}_i)^T$, where $\Phi(\mathbf{X}_i) = (\phi(\mathbf{x}_{i1}), ..., \phi(\mathbf{x}_{in_i}))$ and $\mathbf{1}_{n_i}$ is a $n_i \times n_i$ matrix with all elements equal to $1/n_i$. Then, $tr(\mathbf{\Sigma}_i^\Phi \mathbf{\Sigma}_k^\Phi) = tr(\Phi(\mathbf{X}_i)(\mathbf{I} - \mathbf{1}_{n_i})\Phi(\mathbf{X}_i)^T \Phi(\mathbf{X}_k)(\mathbf{I} - \mathbf{1}_{n_k})\Phi(\mathbf{X}_k)^T) = tr(\mathbf{K}_{ki}(\mathbf{I} - \mathbf{1}_{n_i})\mathbf{K}_{ik}(\mathbf{I} - \mathbf{1}_{n_k}))$, where $\mathbf{K}_{ik} = \Phi(\mathbf{X}_i)^T\Phi(\mathbf{X}_k)$. Let $\tilde{\mathbf{K}}_{ki} = \mathbf{K}_{ki}(\mathbf{I} - \mathbf{1}_{n_i})$ and $\tilde{\mathbf{K}}_{ik} = \mathbf{K}_{ik}(\mathbf{I} - \mathbf{1}_{n_k})$. We can rewrite this result as,

$$tr(\tilde{\mathbf{K}}_{ki}\tilde{\mathbf{K}}_{ik}) = \sum_p \sum_q \tilde{\mathbf{K}}_{ki}^{pq} \tilde{\mathbf{K}}_{ik}^{qp},$$

where $\widetilde{\mathbf{K}}_{ki}^{pq}$ is the $(p,q)th$ entry of $\widetilde{\mathbf{K}}_{ki}$. Denote the partial derivative of an $m \times n$ matrix $\mathcal{K}$ with respect to σ as $\frac{\partial \mathcal{K}}{\partial \sigma} = \left[\frac{\partial \mathcal{K}^{pq}}{\partial \sigma}\right]_{p=1,...,m,q=1,...,n}$, with $\frac{\partial \mathcal{K}^{pq}}{\partial \sigma} = \frac{\partial k(x_p, x_q)}{\partial \sigma} = \frac{\|x_p - x_q\|^2}{\sigma^3} \exp(-\frac{\|x_p - x_q\|^2}{2\sigma^2})$ when using the RBF function. Then,

$$\begin{aligned}
\frac{\partial tr(\mathbf{\Sigma}_i^\Phi \mathbf{\Sigma}_k^\Phi)}{\partial \sigma} &= \frac{\partial tr(\tilde{\mathbf{K}}_{ki}\tilde{\mathbf{K}}_{ik})}{\partial \sigma} \\
&= \sum_p \sum_q \left(\frac{\partial \tilde{\mathbf{K}}_{ki}^{pq}}{\partial \sigma} \tilde{\mathbf{K}}_{ik}^{qp} + \tilde{\mathbf{K}}_{ki}^{pq} \frac{\partial \tilde{\mathbf{K}}_{ik}^{qp}}{\partial \sigma} \right) \\
&= \sum_p \sum_q \left[\left(\frac{\partial \mathbf{K}_{ki}^{pq}}{\partial \sigma} - \frac{\partial \mathbf{K}_{ki}^{pq}}{\partial \sigma} \mathbf{1}_i \right) \tilde{\mathbf{K}}_{ik}^{qp} + \tilde{\mathbf{K}}_{ki}^{pq} \left(\frac{\partial \mathbf{K}_{ik}^{qp}}{\partial \sigma} - \frac{\partial \mathbf{K}_{ik}^{qp}}{\partial \sigma} \mathbf{1}_k \right) \right].
\end{aligned}$$

Next, note that $Q_2(\phi)$ can be written as

$$Q_2(\phi) = \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} p_i p_k d_{ik},$$

where

$$\begin{aligned}
d_{ik} &= (\mu_i^\phi - \mu_k^\phi)^T (\mu_i^\phi - \mu_k^\phi) \\
&= (\Phi(\mathbf{X}_i)\mathbf{1}_i - \Phi(\mathbf{X}_k)\mathbf{1}_k)^T (\Phi(\mathbf{X}_i)\mathbf{1}_i - \Phi(\mathbf{X}_k)\mathbf{1}_k) \\
&= \mathbf{1}_i^T \mathbf{K}_{ii} \mathbf{1}_i - 2\mathbf{1}_i^T \mathbf{K}_{ik} \mathbf{1}_k + \mathbf{1}_k^T \mathbf{K}_{kk} \mathbf{1}_k.
\end{aligned}$$

Using this notation, the gradient of $Q_2(\phi)$ with respect to σ is

$$\frac{\partial Q_2(\phi)}{\partial \sigma} = \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} p_i p_k \frac{\partial d_{ik}}{\partial \sigma} = \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} p_i p_k \left(\mathbf{1}_i^T \frac{\partial \mathbf{K}_{ii}}{\partial \sigma} \mathbf{1}_i - 2\mathbf{1}_i^T \frac{\partial \mathbf{K}_{ik}}{\partial \sigma} \mathbf{1}_k + \mathbf{1}_k^T \frac{\partial \mathbf{K}_{kk}}{\partial \sigma} \mathbf{1}_k \right).$$

This result allows us to iteratively determine an appropriate solution. To see that the solution found with such a gradient descent technique is an appropriate one, recall that Theorem 2 showed Q_1 monotonically increases if $tr(S_B) > tr(S_W)$. In most practical problems this condition is satisfied, since otherwise the classes mostly overlap and the classification problem is not solvable (i.e., there is a very large classification error in the original feature space). This means there is an identifiable global maximum. We now note that the same applies to Q_2. That is, as long as the class distributions do not overlap significantly, Q_2 has a unique maximum for a sigma value in between the averaged within class sample distances and the averaged between class sample distances. To see this, note that for every Q_2 calculated for a pair of classes (i.e., classes 1 and 2), there are three main components: the sum of the kernel matrix elements in class 1, in class 2, and between classes 1 and 2. Each of these components monotonically increases with respect to sigma (starting with $1/n_1$, $1/n_2$, 0, and converging to 1). The fastest increases occur for sigma around the averaged distance in that component; e.g., for within class 1, this will be around the averaged distance of the samples in that class. This means that the within class components will converge earlier than the between class distances. Hence, the sum of the within class subtracted with

TABLE S1

RECOGNITION RATES (IN PERCENTAGES) WITH NEAREST MEAN

Data set	ksda$_H$	ksda$_F$	ksda$_B$	ksda$_{CV}$	kda$_H$	kda$_F$	kda$_B$	kda$_{CV}$	knda$_H$	knda$_F$	knda$_B$	knda$_{CV}$
ETH-80	**82.6***	73.5	61.7	77.4	**82.6***	81.6	61.7	71.6	**76.2**	74.6	65.6	73.6
AR database	**88.1***	78.2	65.5	84.2	**87.5***	86.7	69.5	84.2	71.3	61.4	72.5	**74.3**
SPDM	**84.6***	80.1	67.9	**83.9***	**84.6***	83.2	67.9	83.3	**82.4**	**82.9**	53.4	75.6
Monk1	**88.2***	85.0	71.1	**88.0***	84.0	**89.6***	65.3	83.1	**70.1**	65.7	50.0	63.4
Monk2	76.6	**82.2***	56.7	74.5	**80.1**	75.2	55.6	70.1	**73.5**	64.8	61.8	71.8
Monk3	**96.3***	88.7	85.4	94.0	**93.1**	89.7	85.7	82.4	67.6	63.7	**77.8**	66.4
Ionosphere	93.4	84.8	88.1	**96.0***	**93.4**	86.1	67.6	80.8	74.8	62.3	65.6	**78.2**
Pima	**80.4***	77.4	70.2	**80.4***	**78.6**	75.0	75.0	72.6	65.5	67.3	**70.8**	66.7
Mnist	**98.0***	96.9	92.0	97.4	**98.1***	96.6	92.0	97.2	94.6	94.3	93.1	**96.4**
Rank	1.9*	7.0	13.3	3.6	2.8	5.4	14.2	9.2	12.2	14.7	15.8	13.3

Data set	mog	ksvm	kpca	pca	lda	nda	apac	hlda	lpp	rda	sda
ETH-80	69.2	81.8	56.9	56.5	63.3	64.9	64.0	58.2	65.9	71.6	70.9
AR database	75.5	86.7	42.2	24.0	79.3	69.7	24.2	67.4	46.2	78.6	79.3
SPDM	73.4	84.7*	62.6	66.4	44.5	52.5	65.3	68.0	54.7	59.5	69.3
Monk1	80.3	83.6	67.4	66.0	64.6	64.8	66.0	66.2	44.4	72.0	66.7
Monk2	75.9	82.6	53.7	53.5	55.1	60.0	53.5	53.5	48.6	60.0	55.1
Monk3	89.4	93.5	78.9	80.6	63.9	81.3	80.6	81.3	75.5	86.3	80.8
Ionosphere	82.1	96.0	89.4	62.3	57.0	92.1	62.3	90.1	55.0	82.8	90.1
Pima	75.0	79.2	50.0	56.0	61.3	74.4	56.0	77.4	67.9	66.7	61.3
Mnist	88.6	97.6*	80.6	82.2	86.7	85.9	82.2	85.5	80.1	87.0	88.2
Rank	9.8	2.7	18.0	19.1	18.3	14.4	18.4	14.1	19.9	12.4	12.8

TABLE S2

RECOGNITION RATES (%) WITH NEAREST NEIGHBOR

Data set	ksda$_H$	ksda$_F$	ksda$_B$	ksda$_{CV}$	kda$_H$	kda$_F$	kda$_B$	kda$_{CV}$	knda$_H$	knda$_F$	knda$_B$	knda$_{CV}$
ETH-80	**82.8***	73.6	62.3	76.8	**82.8***	81.0	62.3	71.6	**76.2**	74.6	68.0	70.6
AR database	**96.7***	78.3	66.9	84.2	**88.3**	87.5	71.3	84.2	**69.2**	64.2	**70.6**	**70.2**
SPDM	**84.9***	80.1	68.2	83.7	**84.9***	**84.2**	68.2	83.3	73.9	**75.6**	33.5	70.3
Monk1	**89.1***	84.5	78.2	87.5	84.3	**89.6***	72.5	83.1	**78.2**	77.1	74.5	72.2
Monk2	77.8	**83.1**	86.1	75.7	**80.1**	75.2	77.6	70.1	**85.0***	81.0	79.9	78.5
Monk3	**94.4***	87.7	81.5	89.8	**93.5**	88.0	89.4	82.4	**82.1**	81.3	77.6	80.3
Ionosphere	**94.4**	84.8	91.4	**94.0**	**94.4**	86.5	70.9	80.8	87.4	86.1	**90.1**	86.1
Pima	75.0	73.8	66.7	**76.8**	70.2	69.8	64.9	**72.6**	67.3	67.3	66.1	**69.1**
Mnist	**97.8***	96.9	91.8	97.2	**97.2**	**97.1**	**91.8**	96.7	**95.6**	**95.4**	92.1	**95.5**
Rank	2.9*	8.0	13.6	5.3	3.7	7.7	15.4	10.8	11.3	12.7	15.7	14.1

Data set	mog	ksvm	kpca	pca	lda	nda	apac	hlda	lpp	rda	sda
ETH-80	69.2	81.8	62.2	64.3	64.3	59.8	73.6	56.5	63.6	71.6	70.6
AR database	75.5	86.7	42.5	58.6	77.7	77.0	59.1	67.5	41.8	78.6	77.7
SPDM	73.4	84.7	75.0	81.5	66.5	48.8	81.1	65.3	54.1	59.5	66.1
Monk1	80.3	83.6	90.3*	81.3	69.0	68.3	81.0	84.2	61.6	72.0	75.7
Monk2	75.9	82.6	68.3	66.7	67.4	82.6	79.6	83.6	82.4	60.0	67.4
Monk3	89.4	93.5	87.8	87.3	70.6	83.6	88.4	84.5	80.6	86.3	85.9
Ionosphere	82.1	96.0*	89.4	92.1	74.8	88.8	92.1	88.7	68.2	82.8	93.4
Pima	75.0	79.2	56.0	64.3	57.7	69.1	62.5	68.5	66.8	66.7	57.7
Mnist	88.6	97.6	94.1	90.1	89.7	85.6	89.3	80.6	96.0	87.0	93.7
Rank	12.6	3.2	14.2	14.3	18.4	15.4	12.7	14.3	17.4	16.2	14.1

two times the between class elements (in the kernel matrix) will result in a maximum in between the averaged within class sample distances and between class sample distances.

In some applications where our conditions may not hold, it would be appropriate to test a few starting values to determine the best solution. We did not require this procedure in our experiments.

II. MIXTURE OF GAUSSIANS (MOG) APPROACH

Mixture Models is a widely used approach to estimating the underlying distribution of the data. In the MoG approach, each class is modeled as a mixture of Gaussians, and the means and covariance matrices are estimated using the Expectation-Maximization (EM) algorithm [1]. The algorithm also needs to determine the number of mixtures. However, as the number of clusters increases, the likelihood of the data also increases and applying the EM algorithm would lead to the "optimal" representation of one Gaussian per sample. This is an overfitting problem. A classical way to resolve this is using the Minimum Description Length (MDL) criterion [4], defined as

$$MDL(H,\theta) = -logp(\mathbf{X}|H,\theta) + \frac{1}{2}Llog(N), \tag{1}$$

TABLE S3

RECOGNITION RATES (%) WITH THE SMOOTH NEAREST-NEIGHBOR CLASSIFIER

Data set	ksda$_H$	ksda$_F$	ksda$_B$	ksda$_{CV}$	kda$_H$	kda$_F$	kda$_B$	kda$_{CV}$	knda$_H$	knda$_F$	knda$_B$	knda$_{CV}$
ETH-80	**83.5***	73.9	62.3	76.4	**83.5***	82.8	62.3	72.9	**76.2**	74.2	68.2	71.2
AR database	**96.6***	78.5	66.9	85.1	**90.6**	86.7	71.3	85.1	70.9	63.2	70.6	**72.6**
SPDM	**84.3***	75.3	68.2	**83.9***	**84.3***	83.4	68.2	82.6	75.6	**77.9**	35.6	71.5
Monk1	**90.2***	76.6	71.5	82.9	**89.6**	87.7	72.2	88.7	**65.2**	62.0	61.4	62.3
Monk2	**83.3***	77.5	60.6	75.7	80.6	**82.9**	73.8	78.5	**74.1**	64.8	62.3	56.9
Monk3	**94.6***	83.3	86.1	86.3	**93.5**	92.4	89.4	91.2	68.5	64.8	**85.4**	66.2
Ionosphere	**94.3**	84.8	84.8	86.1	**94.3**	86.8	80.1	86.8	80.8	**82.8**	77.5	78.1
Pima	**80.4***	76.8	79.2	76.2	**78.6**	73.0	64.9	69.0	**72.0**	67.9	69.0	67.9
Mnist	**97.8***	96.9	91.8	97.3	**97.2**	**97.2**	91.8	96.7	**95.6**	95.4	92.1	**95.6**
Rank	1.2*	9.4	14.4	6.7	2.7	4.6	15	6.9	14.2	14.7	17.6	16.1

Data set	mog	ksvm	kpca	pca	lda	nda	apac	hlda	lpp	rda	sda
ETH-80	69.2	81.8	60.3	67.1	64.3	63.5	71.2	59.1	64.3	71.6	72.3
AR database	75.5	86.7	49.5	44.5	70.9	77.3	60.2	67.5	35.5	78.6	70.9
SPDM	73.4	84.7	75.1	77.0	56.2	50.2	81.2	53.4	50.2	59.5	69.5
Monk1	80.3	83.6	77.3	78.2	67.4	77.8	69.4	71.5	59.0	72.0	79.2
Monk2	75.9	82.6	58.6	56.7	70.6	70.6	70.4	58.3	72.0	60.0	70.6
Monk3	89.4	93.5	91.2	89.7	70.8	91.9	89.6	93.8	87.0	86.3	90.5
Ionosphere	82.1	96.0*	82.1	82.1	74.8	83.4	91.1	94.0	62.9	82.8	89.4
Pima	75.0	79.2	60.7	70.2	57.7	70.2	63.8	72.6	66.1	66.7	57.7
Mnist	88.6	97.6	94.1	90.1	89.8	86.0	89.4	82.6	96.1	87.0	93.5
Rank	11.6	2.7	15.6	14.6	17.8	13.4	13.9	14.9	18.1	14.9	11.9

TABLE S4

RECOGNITION RATES (%) WITH LINEAR SVM

Data set	ksda$_H$	ksda$_F$	ksda$_B$	ksda$_{CV}$	kda$_H$	kda$_F$	kda$_B$	kda$_{CV}$	knda$_H$	knda$_F$	knda$_B$	knda$_{CV}$
ETH-80	**83.0***	73.6	61.9	77.4	**83.0***	82.2	61.9	71.3	**75.6**	**75.2**	65.6	74.6
AR database	**88.1***	79.6	65.5	83.1	**87.5***	86.7	69.5	83.1	**79.4**	75.7	72.5	78.6
SPDM	82.1	**84.6***	67.5	82.3	82.1	**83.6**	67.5	82.6	82.2	82.9	52.7	**84.0**
Monk1	**89.1***	88.2	50.0	86.1	84.7	**89.7***	52.1	86.1	**69.9**	62.5	50.0	63.4
Monk2	77.1	**81.5**	67.1	73.8	**80.1**	75.2	67.1	75.1	67.1	**83.1***	67.1	67.1
Monk3	**95.6***	91.9	47.2	94.4	**92.8**	89.1	47.2	81.5	**81.7**	**81.7**	47.2	81.0
Ionosphere	93.4	86.1	82.1	**96.7***	**93.4**	86.1	82.1	82.1	**82.1**	**82.1**	**82.1**	**82.1**
Pima	**79.8***	78.6	64.9	**79.8***	**78.0***	75.0	64.3	**72.8**	**64.3**	**64.3**	**64.3**	**64.3**
Mnist	**97.9**	96.9	92.0	97.3	**98.1***	96.7	92.0	97.2	94.7	94.3	93.3	**96.2**
Rank	2.8*	5.6	17.8	4.3	4.1	5.8	17.7	9.5	11.9	11.6	17.3	13.0

Data set	mog	ksvm	kpca	pca	lda	nda	apac	hlda	lpp	rda	sda
ETH-80	69.2	81.8	65.3	60.1	65.3	61.8	68.4	68.4	62.1	71.6	67.8
AR database	75.5	86.7	42.1	66.7	79.3	69.7	67.2	70.1	44.2	78.6	79.3
SPDM	73.4	84.7*	66.7	76.5	50.3	49.0	82.1	69.3	45.5	59.5	69.0
Monk1	80.3	83.6	88.4*	67.8	65.6	66.4	67.8	68.5	44.9	72.0	66.7
Monk2	75.9	82.6	50.0	67.1	67.1	67.5	65.6	67.1	67.1	60.0	67.1
Monk3	89.4	93.5	94.4	81.3	63.9	83.3	80.6	81.9	78.5	86.3	84.7
Ionosphere	82.1	96.0	82.1	84.8	84.8	88.1	93.4	93.4	82.1	82.8	90.1
Pima	75.0	79.2	64.3	68.6	64.9	76.8	77.4	76.2	76.2	66.7	64.9
Mnist	88.6	97.6	81.0	82.2	86.9	85.9	83.1	85.4	80.1	87.0	88.2
Rank	11.5	3.3	16.1	16.1	15.8	14.6	13.6	12.5	19.0	13.8	12.9

where $logp(\mathbf{X}|H,\theta)$ is the log-likelihood of the mixture model, $\mathbf{X}$ is the data matrix, H is the number of clusters, and θ are the parameters of the model. $\frac{1}{2}Llog(N)$ is a penalty function, with $L = H-1+H(D+(D+1)D/2)$, where N is the number of samples, and D is the dimension of the data. The minimization of the MDL criterion thus determines the number of mixtures in our model.

After the parameters of the MoG are estimated, the Maximum Likelihood (ML) rule is used to define the nonlinear classification boundary. Note that, as in SVM, this classification is directly done in the original space.

III. EXPERIMENTAL RESULTS: DETAILS

In the main paper we provided a summary of the experimental results. We now present a slightly larger comparison – against additional method. These comparisons are given in Tables S1-S4. These tables provide comparative results against all the nonlinear and linear approaches described in the paper. As with the results given in the main manuscript, we see that the proposed Homoscedastic criterion consistently selects those kernel parameters which yield the highest classification accuracies. The bolded values are given to identify which of the four criteria used for optimizing the kernel DA approaches yields the best results. For example, in Table S1, when using KSDA, the homoscedastic criterion derived in this paper yields better results than those obtained with the Fisher,

Bregman and CV criteria 7 out of 9 times. In comparison, the Fisher criterion only yields the best results once, and CV 4 times. We are also interested in which algorithm, over all those listed in the table, yields the top result. That is, which approach is to be preferred. The top results (with statistical significance) are marked with an asterisk in the tables.

We now provide a quantitative analysis to rank the different approaches tested. This requires that we provide an analytical study of the results of multiple algorithms on multiple databases. To do this we follow the approach of [2]. In particular, we used the Friedman's statistical test of significance. The null hypothesis is that all algorithms are equally good and that the numerical differences are given by noise. The resulting rankings are given in Tables S1-S4, with statistical significance marked with an asterisk. We see that KSDA$_H$ consistently yields the best classification accuracies, regardless of the classifier used in the reduced space. KDA$_H$ and KSVM rank second. We also see that slightly better results are given when the smooth nearest-neighbor classifier of [3] is used in the subspace of KSDA$_H$; followed by the nearest mean. This suggests the subclasses are close to homoscedastic and that our goal has been achieved.

The differences in running time observed in Table V between KSDA and KSVM are given by implementation or running details rather than algorithm complexity. Both algorithms share the same complexity and will generally be equivalent with regard to the training time. KSVM does, however, considerably reduce the size of the feature spaces. KDA$_H$ is about an order of magnitude faster than KSVM and yields comparable results to it – as demonstrated by the rankings generated above.

REFERENCES

[1] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal Royal Statistical Society 30(1):1-38, 1977.
[2] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine Learning Research, 7:1-30, 2006.
[3] O. Pujol and D. Masip, "Geometry-based ensembles: Towards a structural characterization of the classification boundary," IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(6):1140-1146, 2009.
[4] J. Rissanen, "Universal coding, information, prediction, and estimation," IEEE Trans. Inform. Theory, 30:629-636, 1984.