Combining classifiers based on kernel density estimates and Gaussian mixtures

Edgar Acuña

Department of Mathematics
University of Puerto Rico
Mayagüez Campus
(www.math.uprm.edu/~edgar)

**This research is supported by Grant N00014-00-1-0360 from ONR*

OUTLINE

- The supervised classification problem
- The misclassification error
- Combining classifiers
- Kernel density estimators classifiers
- Feature selection problem
- Gaussian mixtures classifiers
- Results and concluding remarks
- Current work

The Misclassification Error

Let C(x, L) be the classifier constructed by using the training sample L., and T another large sample from the same population as L was drawn from, then the misclassification error (ME) of the classifier C is the proportion of misclassified cases of T using C.

The ME can be descomposed as

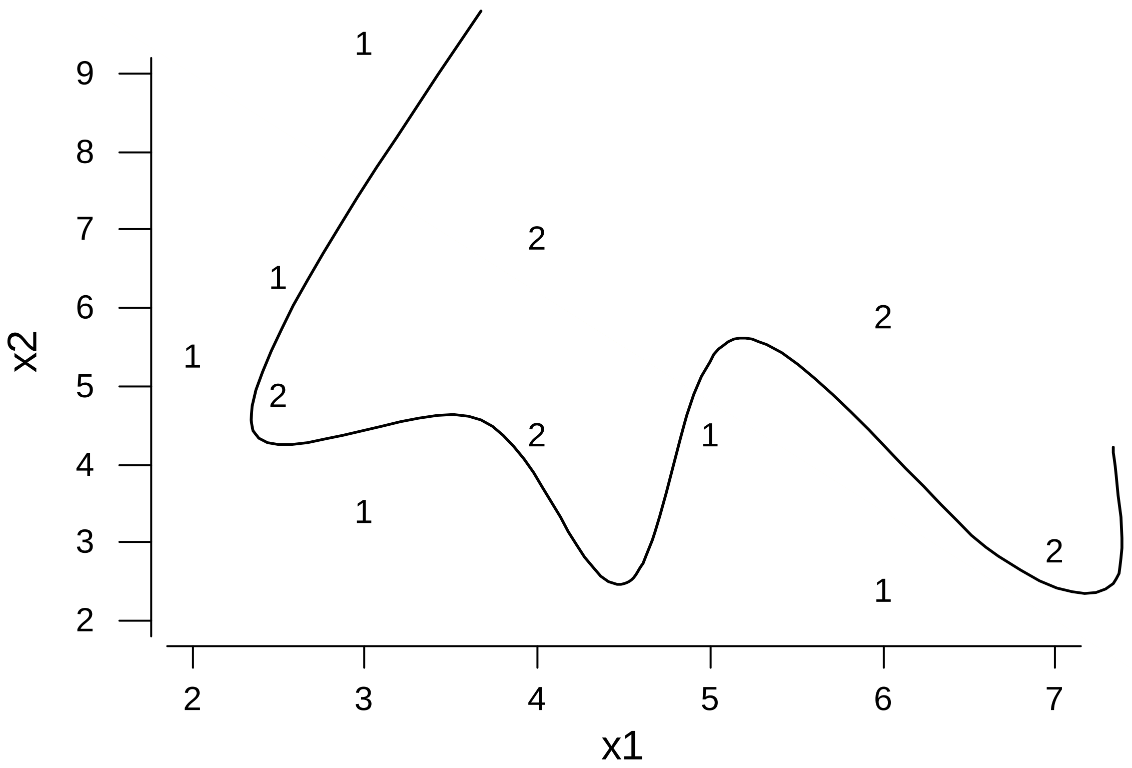
$$ME(C)=ME(C^*)+Bias^2(C) +Var(C)$$

where $C^*(x)=\text{argmax}_j P(Y=j/X=x)$ (Bayes Classifier)

Methods to estimate ME: Resubstitution, Crossvalidation, Bootstrapping

The classifier may either overfit the data (Low bias and large variance) or underfit the data (Large bias and small variance).
Breiman (1996) heuristically defines a classifier as unstable if a small change in the data *L* can make large changes in the classification. Unstable classifiers have low bias but high variance.
CART and Neural networks are unstable classifiers.
Linear discriminant analysis and K-nearest neighbor classifiers are stable.

Overfitting



Combining classifiers

Combining the predictions of several classifiers the variance and bias could be reduced. This combination is called an **Ensemble** and in general is more accurate than the individual classifiers.

Methods for creating ensembles are: **Bagging** (Bootstrap aggregating by Breiman, 1996)

AdaBoosting (Adaptive Boosting by Freund and Schapire, 1996)

Arcing (Adaptively resampling and combining, by Breiman (1998).

The Bagging Algorithm: Breiman (1996)

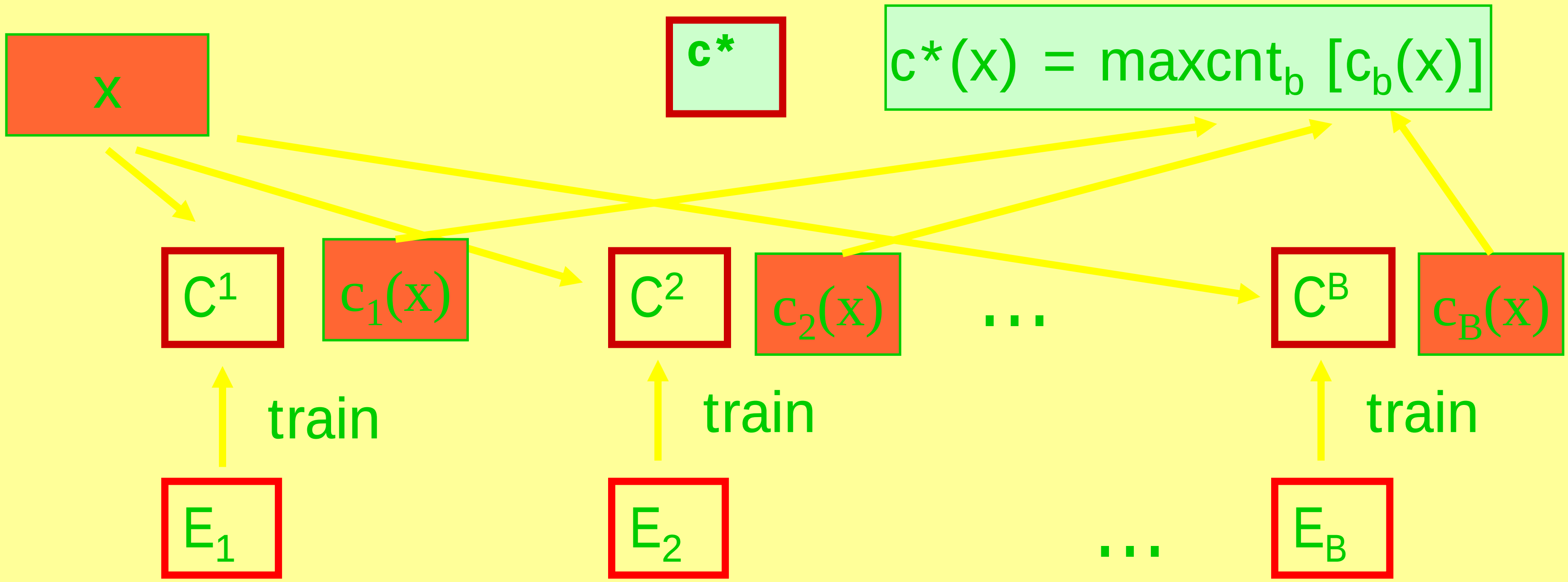
Input: learning set L , classifier C, integer T (number of bootstrap samples)

1. For i=1 to T {
2. B_i= Bootstrap sample from L (i.i.d. sample with replacement)
3. $C_i=C(B_i)$
4. }
5. $C_A(x)= \arg\max_{j\in\{1,\ldots,J\}} \sum_{i:C_i(x)=j} 1$ (The class most voted)

Output: Ensemble C_A

Bagging

- From the training sample **L** select B random samples with replacement (bootstrap samples) obtaining **B** different traioning samples $L_1,...,L_B$ of size N.
- For each sample L_b a classifier C^b is built .
- Using 10-fold cross validation , each case x of E is assigned to the class c*(x)=j by voting.



Adaboosting Algorithm:Freund & Schapire[1996]

Input: Learning set L , classifier C, integers N, T

1. B= L with weights $w_1(x_j)=1/N$. For j=1,…..N

2. For i=1 to T { $C_i=C(B)$

3. Set $e_i = \sum_{x_j \in B: C_i(x_j) \neq y_j} w_i(\mathbf{x}_j)$. If either $e_i>1/2$ or $e_i=0$ then restart assigning equal weights.

4. Set $\beta_i=e_i/(1-e_i)$

5. Udpate the weights: for each $x_j \in B$, if $C_i(x_j) \neq y_j$ then $w_{i+1}(x_j)=w_i(x_j)/2e_i$, else $w_{i+1}(x_j)=w_i(x_j)/2(1-e_i)$)}

6. $C^*(x)=\arg\max_{j\in\{1,\ldots J\}} \sum_{i:C_i(x)=j} \log\frac{1}{\beta_i}$

Output: Ensemble C*

Previous results on combining classifiers

Reference	Classifier	Relative Improv. (%)	
		Bagging	**Boosting**
Breiman (1996)	**CART**	**29.0**	**------**
Freund & Schapire (1996)	**C4.5**	**20.0**	**24.8**
Quinlan (1996)	**C4.5**	**10.0**	**15.01**
Maclin & Opitz (1997)	**C4.5**	**18.5**	**22.0**
Maclin & Opitz (1997)	**Neural Net**	**13.3**	**17.1**
Breiman (1998)	**CART**	**36.0**	**48.4**
Bauer & Kohavi (1999)	**MC4**	**14.5**	**27.0**
Dietterich(2000)	**C4.5**	**16.9**	**22.4**
Daza (2002)	**G.M.**	**10.1**	**0.8**
Acuna & Rojas (2002)	**Kernel**	**4.9**	**1.9**

- Bagging reduces variance. AdaBoosting reduces both bias and variance
- Bagging can be parallelized easily, but Boosting is essentially sequential and only some part of the algoruthm can be parallelized.
- Boosting is only useful for large sample datasets and for classifiers that perform poorly.

Bayesian approach to classification

An object with measurement vector x is assigned to the class j* if

$$P(Y=j^*/x)>P(Y=j/x) \text{ for all } j \neq j^*$$

f

By Bayes's theorem $P(Y=j/x)=\pi_j f(x/j)/f(x)$

$\pi_j=P(Y=j)$: Prior of the j-th class

$f(x/j)$: Class conditional density

$f(x)$: Density function of x

Thus, $j^* = \text{argmax}_j \, \pi_j f(x/j)$.

Density estimators

- Histograms

$$\hat{f}(x) = \frac{k}{nV}$$

For x in V

- K-nearest neighbors:

$$\hat{f}(\mathbf{x}) = \frac{k}{nV_k}$$

V_k is the volumen including the k nearest neighbors

- Kernel density estimators

Experimental Methodology

- Each dataset is randomly divided in 10 parts. The first of this part is taking as the test sample and the remaining ones as the training sample. Next, 50 bootstrapped samples are taking from the training sample and a KDE classifier is constructed with each of them. Finally each instance of the test sample is assigned to a class by voting using the 50 classifiers. The procedure is repeated with each part and then the whole experiment is repeated 10 times.

Datasets

Dataset			*Features*			
	Instances	*Classes*	*C*	*B*	*N*	*O*
Iris	150	3	4	-	-	-
Sonar	208	2	60			
Glass	214	6	9	-	-	-
Heart-c	303	2	5	3	3	2
Bupa	345	2	6	-	-	-
Ionosphere	351	2	34			
Crx	690	2	6	4	5	-
Breast-w	699	2	9			
Diabetes	768	2	8	-	-	-
Vehicle	846	4	18			
German	1000	2	7	2	10	1
Segment	2310	7	19	-	-	-

Source: UCI Machine Learning Depository

Statistical properties of datasets

Dataset	*Normality*	*Correlation*	*Outliers*
Iris	Yes	Some	Few
Sonar	No	Low	Plenty
Glass	Some	Low	Few
Heart-c	Some	None	Few
Bupa	Some	Low	Some
Ionosphere	No	High	Plenty
Crx	No	None	Plenty
Breast-w	No	None	Plenty
Diabetes	No	Low	Plenty
Vehicle	No	High	Some
German	Some	None	Plenty
Segment	No	High	Plenty

Bagging Performance

Dataset	*Classical Kernel*			*Adaptive Kernel*		
	Single	*Bagged*	*Ratio*	*Single*	*Bagged*	*Ratio*
Iris	3.53	3.60	1.018	4.47	4.26	0.953
Sonar	17.18	17.21	1.001	16.37	15.70	0.959
Glass	44.57	43.83	0.983	35.46	35.51	1.001
Heart-C	22.30	20.80	0.932	22.55	20.13	0.892
Bupa	40.75	40.61	0.996	37.51	37.45	0.998
Ionosphere	10.93	10.48	0.958	10.33	10.08	0.975
Crx	18.93	17.72	0.936	17.83	16.38	0.919
Breast-w	3.64	3.77	1.036	3.94	3.80	0.964
Diabetes	26.38	26.21	0.993	26.26	25.72	0.979
Vehicle	35.15	34.61	0.984	36.74	33.90	0.918
German	28.71	27.95	0.973	27.34	25.12	0.863
Segment	15.86	15.32	0.965	13.41	13.32	0.993
MEAN			0.981			0.951

Boosting Performance

Dataset	*Classical Kernel*			*Adaptive Kernel*		
	Single	*Boosted*	*Ratio*	*Single*	*Boosted*	*Ratio*
Iris	3.53	4.93	1.396	4.47	5.00	1.118
Sonar	17.30	15.77	0.911	16.87	15.57	0.922
Glass	44.11	33.60	0.761	35.14	30.09	0.856
Heart-C	22.56	23.87	1.058	22.34	22.72	1.017
Bupa	40.75	37.79	0.927	37.51	36.78	0.980
Ionosphere	10.91	6.69	0.613	10.42	7.72	0.740
Crx	18.78	19.25	1.024	18.87	18.94	1.003
Breast-w	3.55	4.62	1.301	3.94	5.03	1.276
Diabetes	26.35	31.57	1.198	26.17	30.53	1.166
Vehicle	35.01	29.87	0.853	36.70	29.85	0.813
German	34.84	34.03	0.976	34.91	33.07	0.947
Segment	15.91	4.95	0.311	13.33	4.77	0.357
MEAN			0.944			0.933

Finite mixtures

Let Y=(Y_1, Y_2, Y_3,..., Y_n), be a random sample of size n where Y_j is a random vector p-dimensional with density function $f\left(y_j;\psi\right)$

La función de densidad de Y_i, $f\left(y_j;\psi\right)$ puede ser escrita de la forma:

$$f\left(y_j;\psi\right)=\sum_{i=1}^{k}\pi_i f_i\left(y_j;\theta_i\right)$$

$$\pi_i > 0, \quad i = 1, 2, 3, \cdots, k$$

$$\sum_{i=1}^{k}\pi_i = 1$$

Donde las $f_i(y_j;\theta_i)$ son funciones de densidad y se denominan componentes de la mezcla, θ_i el es vector de parámetros desconocidos .

Gaussian Mixture

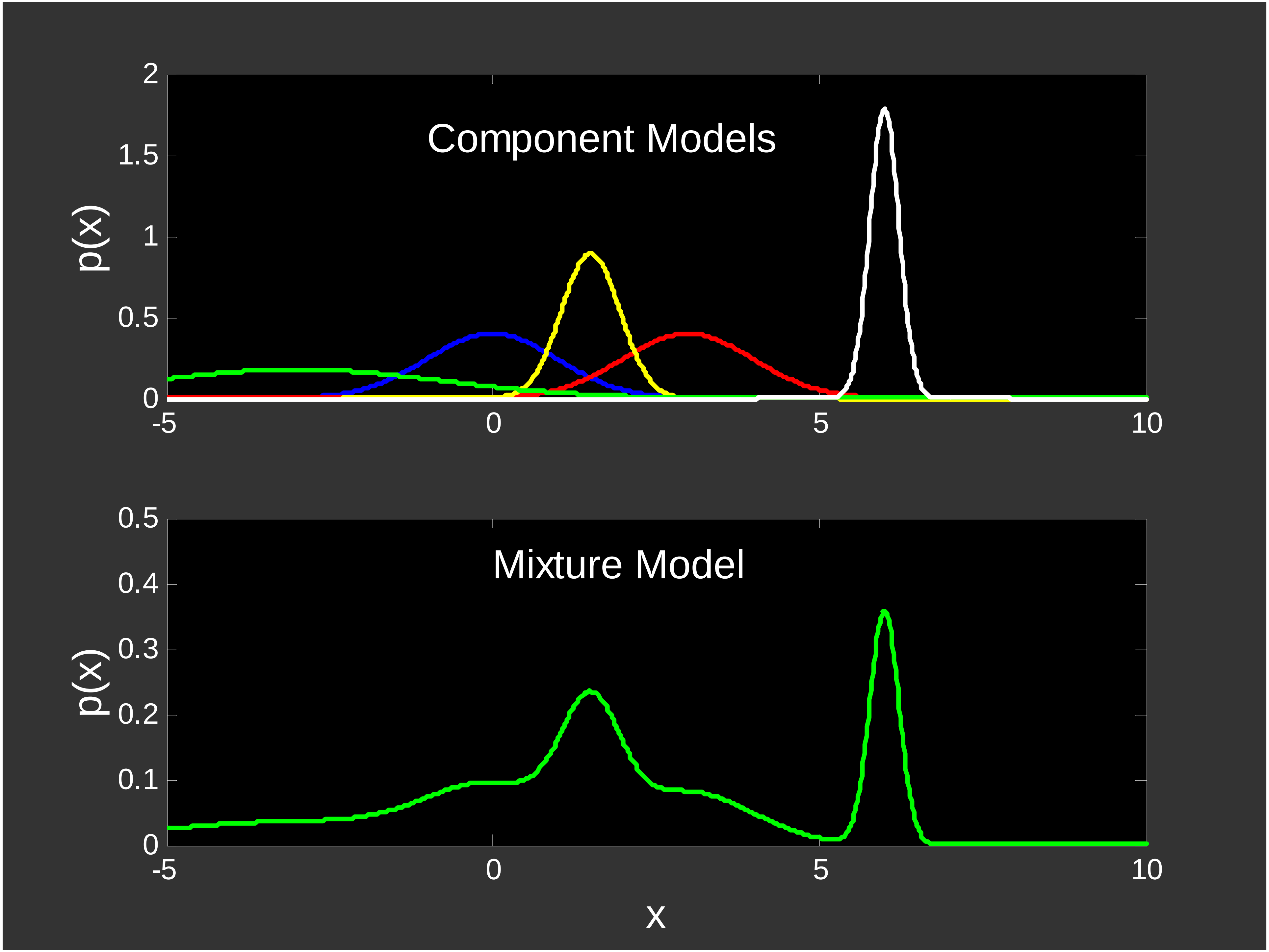
In particular a mixture can have as components Gaussian densities functions. That is

$$f\left(y_j;\psi\right)=\sum_{i=1}^{k}\pi_i f_i\left(y_j;\mu_i,\Sigma_i\right) \qquad (2)$$

where

$$f\left(y_i;\mu_i,\Sigma_i\right)=\left(2\Pi\right)^{-\frac{p}{2}}\left|\Sigma_i\right|^{-1/2}\exp\left[-\frac{1}{2}\left(x-\mu_i\right)^t\Sigma_i^{-1}\left(x-\mu_i\right)\right]$$

The density function (2) is called a Gaussian mixture.


Component Models
2
1.5
1
0.5
0
p(x)
-5
0
5
10
Mixture Model
0.5
0.4
0.3
0.2
0.1
0
p(x)
-5
0
5
10
x

Classification using Gaussian Mixtures

The Gaussian mixture model for the j-th class has density function

$$P(X / y = j) = \sum_{r=1}^{R_j} \pi_{jr} \phi(X; \mu_{jr}, \Sigma)$$

This model has R_j components for the j-th class and the same covariance matrix. The the posterior probability of belonging to the j-th class is given by:

$$P(y = j \backslash X = x) = \frac{\sum_{r=1}^{R_j} \pi_{jr} \phi(X; \mu_{jr}, \Sigma) \Pi_j}{\sum_{l=1}^{j} \sum_{r=1}^{R_l} \pi_{lr} \phi(X; \mu_{lr}, \Sigma) \Pi_l}$$

Parameter Estimation

- Like in the LDA case, the parameter estimation is using Maximun Likelihood. The log-likelihood based on P(y, X) is

$$\sum_{l=1}^{j} \sum_{g_i = j} \log \left(\sum_{r=1}^{R_j} \pi_{lr} \phi(X; \mu_{lr}, \Sigma) \Pi_l \right)$$

- The EM algorithm is used to maximize the log-likelihood

Previous results on GM classifiers

- Hastie and Tibshirani (1994). Discriminant analysis by Gaussian mixtures
- Ormoneit and Tresp (1995)

Resultados experimentales para el conjunto de datos German

	Subclases			
Rep	**2**	**3**	**4**	**5**
1	25.30%	24.10%	23.90%	23.60%
2	25.00%	23.50%	24.00%	25.90%
3	24.30%	24.40%	23.60%	24.30%
4	24.80%	23.80%	23.90%	23.10%
5	24.00%	24.40%	24.80%	23.80%
6	24.20%	23.50%	23.90%	23.90%
7	24.90%	24.20%	23.90%	23.10%
8	24.20%	24.20%	24.40%	24.80%
9	24.70%	22.90%	24.40%	25.30%
10	24.80%	23.10%	25.10%	25.50%
11	23.50%	24.40%	23.50%	24.10%
12	24.80%	24.40%	23.70%	25.20%
13	24.60%	23.10%	24.70%	25.40%
14	25.00%	23.00%	24.20%	23.70%
15	24.20%	24.30%	23.50%	24.80%
Single	**24.62%**	**23.81%**	**24.19%**	**24.33%**
BAGG	**24.60%**	**22.90%**	**23.20%**	**23.40%**
	23.80%	**23.60%**	**23.50%**	**24.00%**
	24.20%	**23.30%**	**23.80%**	**23.70%**
P.Bagg	**24.20%**	**23.27%**	**23.50%**	**23.70%**
Ratio	**0.98**	**0.98**	**0.97**	**0.97**

dataset	subclasses	single	Bagged	ratio
iris	2	2.33	2.00	0.858
sonar	3	24.24	18.90	0.780
heart-c	5	18.46	16.57	0.898
Bupa	5	32.20	30.65	0.952
ionosfera	3	15.32	15.28	0.997
crx	3	13.69	13.17	0.962
Breast-w	3	4.30	3.70	0.860
Diabetes	5	25.50	24.09	0.945
Vehicle	4	20.18	17.53	0.869
German	5	24.33	23.7	0.974
Segment	6	7.19	5.71	0.794
MEAN				0.899

dataset	subclasses	single	Boosting	ratio
iris	2	2.33	2.00	0.858
sonar	3	24.24	20.86	0.861
heart-c	4	17.83	19.53	1.095
Bupa	2	33.07	32.93	0.996
ionosfera	3	15.32	16.07	1.049
crx	2	13.48	13.94	1.034
Breast-w	5	4.57	4.33	0.947
Diabetes	5	25.50	24.97	0.979
Vehicle	4	20.18	20.71	1.026
German	2	24.62	26.07	1.059
Segment	6	7.19	7.25	1.008
MEAN				0.992

Concluding Remarks

- Increasing the number of bootstrapped samples for Bagging seems to improves the misclassification error for both types of classifiers.
- Before feature selection, the adaptive kernel performs better than the standard kernel, but it requires at least three times more computing time.
- After feature selection the performance of bagging deteriorates for both type of kernels.
- Feature selection does a good job, because after that KDE classifiers gives lower ME saving computing time.

Future work

- Analyze the effect of Bagging and Boosting on the bias-variance decomposition of the misclassification error for KDE classifiers.
- Implementation of parallel computer algorithms to build ensembles based on KDE.
- Implementation of parallel computer algorithms to build ensembles based on Gaussian Mixtures.