

# STA 314: Statistical Methods for Machine Learning I

## Lecture 9 - Multi-class Logistic Regression, Discriminant Analysis

Xin Bing

Department of Statistical Sciences  
University of Toronto

In the last lecture, we have learned the logistic regression for binary classification with  $Y \in \{0, 1\}$ .

- Estimating the Bayes rule at any observation  $x \in \mathbb{R}^p$  is equivalent to estimate the conditional probability  $\mathbb{P}(Y = 1 \mid X = x)$ .
- Logistic regression parametrizes the conditional probability by

$$\mathbb{P}(Y = 1 \mid X = x) = \frac{e^{\beta_0 + x^\top \beta}}{1 + e^{\beta_0 + x^\top \beta}}.$$

- We estimate the coefficients by using MLE which can be solved by (stochastic) gradient descent.

# Extension to multi-class classification

When  $Y \in \{0, 1, \dots, K\}$  for  $K \geq 2$ , we need to estimate

$$p_k(x) := \mathbb{P}(Y = k \mid X = x), \quad \forall 1 \leq k \leq K.$$

We assume

$$\begin{aligned} \log \left( \frac{p_1(x)}{p_0(x)} \right) &= \beta_0^{(1)} + \beta_1^{(1)} x_1 + \dots + \beta_p^{(1)} x_p \\ \log \left( \frac{p_2(x)}{p_0(x)} \right) &= \beta_0^{(2)} + \beta_1^{(2)} x_1 + \dots + \beta_p^{(2)} x_p \\ &\vdots \\ \log \left( \frac{p_K(x)}{p_0(x)} \right) &= \beta_0^{(K)} + \beta_1^{(K)} x_1 + \dots + \beta_p^{(K)} x_p \end{aligned}$$

Choice of the baseline is arbitrary.

Equivalently,

$$p_0(x) = \frac{1}{1 + \sum_{k=1}^K e^{x^\top \beta^{(k)}}},$$
$$p_k(x) = \frac{e^{x^\top \beta^{(k)}}}{1 + \sum_{k=1}^K e^{x^\top \beta^{(k)}}}, \quad \forall 1 \leq k \leq K.$$

So classification can be done immediately once  $\beta^{(k)}$ 's are estimated,

# How to estimate coefficients?

A naive approach: separate binary logistic regressions

$$\log \left( \frac{p_k(x)}{p_0(x)} \right) = \beta_0^{(k)} + \beta_1^{(k)} x_1 + \dots + \beta_p^{(k)} x_p$$

Split the data into  $\mathcal{D}_{(1)}^{train}, \dots, \mathcal{D}_{(K)}^{train}$  with  $\mathcal{D}_{(k)}^{train}$  containing all data with  $y \in \{0, k\}$ .

1. For each  $1 \leq k \leq K$ , use  $\mathcal{D}_{(k)}^{train}$  to perform binary logistic regression to estimate  $\beta^{(k)}$  and estimate

$$\frac{p_k(x)}{p_0(x)}$$

2. Assign class label by comparing

$$1, \frac{p_1(x)}{p_0(x)}, \frac{p_2(x)}{p_0(x)}, \dots, \frac{p_K(x)}{p_0(x)}$$

# Why naive?

- Estimation of  $\beta^{(k)}$ 
  - ▶ only uses  $\mathcal{D}^{train}_{(k)}$ , data points in class  $\{0, k\}$
  - ▶ ignore all data points in other classes
- The label  $1\{y_i = k\}$  is **dependent** on all other  $1\{y_i = k'\}$  for  $k' \neq k$ . Intuitively, this dependence can aid estimation of  $\beta^{(k)}$  by using data from all classes.
- What should we use instead?

# MLE for multi-class logistic regression

For  $(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)$ , the log-likelihood of  $(\beta^{(1)}, \dots, \beta^{(K)})$  is proportional to

$$\begin{aligned} & \sum_{i=1}^n \log \left( \prod_{k=0}^K p_k(\mathbf{x}_i)^{1\{y_i=k\}} \right) \\ &= \sum_{i=1}^n \sum_{k=0}^K 1\{y_i = k\} \log(p_k(\mathbf{x}_i)) \\ &= \sum_{i=1}^n \left[ 1\{y_i = 0\} \log(p_0(\mathbf{x}_i)) + \sum_{k=1}^K 1\{y_i = k\} \log(p_k(\mathbf{x}_i)) \right] \\ &= \sum_{i=1}^n \left[ \sum_{k=1}^K 1\{y_i = k\} \mathbf{x}_i^\top \beta^{(k)} - \sum_{k=0}^K 1\{y_i = k\} \log \left( 1 + \sum_{k=1}^K e^{\mathbf{x}_i^\top \beta^{(k)}} \right) \right] \\ &= \sum_{i=1}^n \left[ \sum_{k=1}^K 1\{y_i = k\} \mathbf{x}_i^\top \beta^{(k)} - \log \left( 1 + \sum_{k=1}^K e^{\mathbf{x}_i^\top \beta^{(k)}} \right) \right] \end{aligned}$$

# Gradient of $\ell(\beta^{(k)})$

For any  $1 \leq k \leq K$ ,

$$\begin{aligned}\frac{\partial \ell(\beta^{(1)}, \dots, \beta^{(K)})}{\partial \beta^{(k)}} &= \sum_{i=1}^n \left[ 1\{y_i = k\} \mathbf{x}_i - \frac{\mathbf{x}_i e^{\mathbf{x}_i^\top \beta^{(k)}}}{1 + \sum_{k=1}^K e^{\mathbf{x}_i^\top \beta^{(k)}}} \right] \\ &= \sum_{i=1}^n \left[ 1\{y_i = k\} - \frac{e^{\mathbf{x}_i^\top \beta^{(k)}}}{1 + \sum_{k=1}^K e^{\mathbf{x}_i^\top \beta^{(k)}}} \right] \mathbf{x}_i.\end{aligned}$$

C.f. for binary case

$$\begin{aligned}\frac{\partial \ell(\beta)}{\partial \beta} &= \sum_{i=1}^n \left[ 1\{y_i = 1\} - \frac{e^{\mathbf{x}_i^\top \beta}}{1 + e^{\mathbf{x}_i^\top \beta}} \right] \mathbf{x}_i \\ &= \sum_{i=1}^n \left[ y_i - \frac{e^{\mathbf{x}_i^\top \beta}}{1 + e^{\mathbf{x}_i^\top \beta}} \right] \mathbf{x}_i.\end{aligned}$$



Therefore, for  $1 \leq k \leq K$ , we update

$$\hat{\beta}_{(t+1)}^{(k)} = \hat{\beta}_{(t)}^{(k)} + \alpha \sum_{i=1}^n \left[ 1\{y_i = k\} - \frac{e^{\mathbf{x}_i^\top \hat{\beta}_{(t)}^{(k)}}}{1 + \sum_{k=1}^K e^{\mathbf{x}_i^\top \hat{\beta}_{(t)}^{(k)}}} \right] \mathbf{x}_i.$$

## Remark:

- the gradient update uses data points from **all classes**!
- better estimation than the naive approach

# An alternative to Logistic Regression

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable<sup>1</sup>.
  - ▶ Discriminant analysis does not suffer from this problem.
- When  $n$  is small and we know more about the data, such as the distribution of  $X \mid Y = k$ 
  - ▶ Discriminant analysis has better performance than the logistic regression model.
- Logistic Regression sometimes does not handle multi-class classification well
  - ▶ Discriminant analysis is more suitable for **multi-class** classification problems.

---

<sup>1</sup>A paper on this.

# Discriminant Analysis

- Logistic regression directly parametrizes

$$\mathbb{P}(Y = k \mid X = x), \quad \forall k \in C.$$

- By contrast, **Discriminant Analysis** parametrizes the distribution of

$$X \mid Y = k, \quad \forall k \in C.$$

Normal distributions are oftentimes used.

# Discriminant Analysis

What does parametrizing  $X \mid Y = k$  buy us?

- By Bayes' theorem,

$$\mathbb{P}(Y = k \mid X = x) = \frac{\mathbb{P}(X = x \mid Y = k)\mathbb{P}(Y = k)}{\mathbb{P}(X = x)}.$$

Thus, to compare two classes  $k \neq k' \in C$ ,

$$\begin{aligned} \mathbb{P}(Y = k \mid X = x) &\geq \mathbb{P}(Y = k' \mid X = x) \\ \iff \mathbb{P}(X = x \mid Y = k)\mathbb{P}(Y = k) &\geq \mathbb{P}(X = x \mid Y = k')\mathbb{P}(Y = k') \end{aligned}$$

# Notation for discriminant analysis

Suppose we have  $K$  classes,  $C = \{0, 1, 2, \dots, K - 1\}$ . For any  $k \in C$ ,

- We write

$$\pi_k := \mathbb{P}(Y = k)$$

as the **prior** probability that a randomly chosen observation comes from the  $k$ th class.

- Write

$$f_k(X) := \mathbb{P}(X = x \mid Y = k)$$

as the **density function** of  $X = x$  from class  $k$ .

- In discriminant analysis, parametric assumption is assumed on  $f_k(X)$ .

# The Bayes rule

- By the Bayes' theorem,

$$p_k(x) := \mathbb{P}(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell \in C} \pi_\ell f_\ell(x)}$$

is called **posterior** probability, i.e. the probability that an observation belongs to the  $k$ th class given its feature.

- According to the Bayes classifier, we should classify a new point  $x$  according to

$$\arg \max_{k \in C} p_k(x) = \arg \max_{k \in C} \frac{\pi_k f_k(x)}{\sum_{\ell \in C} \pi_\ell f_\ell(x)} = \arg \max_{k \in C} \pi_k f_k(x).$$

# Discriminant Analysis for $p = 1$

- Assume that

$$X \mid Y = k \sim N(\mu_k, \sigma_k^2), \quad \forall k \in C,$$

namely,

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2}.$$

- Linear Discriminant Analysis (LDA)** further assumes

$$\sigma_0^2 = \sigma_1^2 = \cdots = \sigma_{K-1}^2 = \sigma^2.$$

# Linear Discriminant Analysis for $p = 1$

- As a result,

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{\ell \in C} \pi_\ell f_\ell(x)} = \frac{\pi_k e^{-\frac{1}{2\sigma^2}(x-\mu_k)^2}}{\sum_{\ell \in C} \pi_\ell e^{-\frac{1}{2\sigma^2}(x-\mu_\ell)^2}}.$$

- The Bayes rule classifies  $X = x$  to

$$\begin{aligned} \arg \max_{k \in C} p_k(x) &= \arg \max_{k \in C} \log(p_k(x)) \\ &= \arg \max_{k \in C} \underbrace{\frac{\mu_k}{\sigma^2} x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k}_{\delta_k(x)} \quad (\text{verify!}) \end{aligned}$$

The name LDA is due to the fact that the **discriminant function**  $\delta_k(x)$  is a linear function in  $x$ .



# Linear Discriminant Analysis for $p = 1$

For binary case, i.e.  $K = 2$ ,

$$\arg \max_{k \in \{0,1\}} p_k(x) = \arg \max_{k \in \{0,1\}} \left[ \frac{\mu_k}{\sigma^2} x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k \right]$$

- If the priors are equal  $\pi_0 = \pi_1$  and suppose  $\mu_1 \geq \mu_0$ , then the Bayes classifier assigns  $X = x$  to

$$\begin{cases} 0 & \text{if } x < \frac{\mu_0 + \mu_1}{2} \\ 1 & \text{if } x \geq \frac{\mu_0 + \mu_1}{2} \end{cases}$$

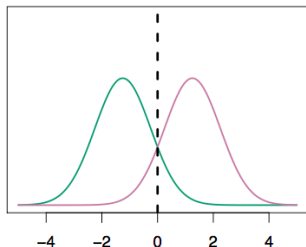
The line  $x = (\mu_0 + \mu_1)/2$  is called [the Bayes decision boundary](#).

# Example of LDA in binary classification

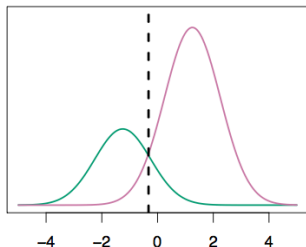
Consider  $\mu_0 = -1.5$ ,  $\mu_1 = 1.5$ , and  $\sigma = 1$ . The curves are  $p_0(x)$  (green) and  $p_1(x)$  (red). The dashed vertical lines are the Bayes decision boundary.

$$f^*(x) = \begin{cases} 0 & \text{if } x < \frac{\mu_0 + \mu_1}{2} = 0 \\ 1 & \text{if } x \geq \frac{\mu_0 + \mu_1}{2} = 0 \end{cases}$$

$\pi_1=.5, \pi_2=.5$



$\pi_1=.3, \pi_2=.7$



# Compute the Bayes classifier

- If we know  $\mu_0, \dots, \mu_{K-1}$ ,  $\sigma^2$  and  $\pi_0, \dots, \pi_{K-1}$ , then we can construct the Bayes rule

$$\arg \max_{k \in C} \delta_k(x) = \arg \max_{k \in C} \left\{ \frac{\mu_k}{\sigma^2} x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k \right\}.$$

- However, we typically don't know these parameters. We need to use the training data to estimate them!

# Estimation under LDA

Given training data  $(x_1, y_1), \dots, (x_n, y_n)$ , for all  $k \in C$ ,

- We have

$$n_k = \sum_{i=1}^n 1\{y_i = k\}.$$

- We estimate  $\pi_k$  by

$$\hat{\pi}_k = \frac{n_k}{n}.$$

- We estimate  $\mu_k$  and  $\sigma^2$  by

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2.$$

These are actually the MLEs.

# The LDA classifier

- We estimate  $\delta_k(x)$  by the plug-in estimator

$$\hat{\delta}_k(x) = \frac{\hat{\mu}_k}{\hat{\sigma}^2}x - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \hat{\pi}_k.$$

- The LDA classifier assigns  $x$  to

$$\arg \max_{k \in C} \hat{\delta}_k(x).$$

- How about the case when  $p > 1$ ?

# Linear Discriminant Analysis for $p > 1$

- Recall that the posterior probability has the form

$$P(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell \in C} \pi_\ell f_\ell(x)},$$

- Now, we assume

$$X \mid Y = k \sim N_p(\mu_k, \Sigma), \quad \forall k \in C,$$

that is,

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}.$$

- The discriminant function becomes

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k.$$

C.f.

$$\delta_k(x) = \frac{\mu_k}{\sigma^2} x - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

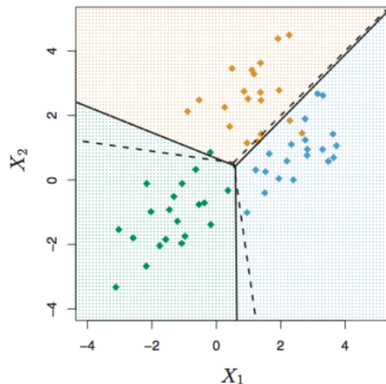
for the univariate case.

- The Bayes decision boundaries are the set of  $x$  for which

$$\delta_k(x) = \delta_\ell(x), \quad \forall k \neq \ell,$$

which are again **linear hyperplanes** in  $x$ .

# Example



There are three classes (orange, green and blue) with two features  $X_1$  and  $X_2$ . Dashed lines are the Bayes decision boundaries. Solid lines are their estimates based on the LDA. 2



# Estimation under LDA for $p > 1$

Given the training data  $(x_1, y_1), \dots, (x_n, y_n)$ , for any  $k \in C$ ,

- We have

$$n_k = \sum_{i=1}^n 1\{y_i = k\}.$$

- We estimate  $\pi_k$  by

$$\hat{\pi}_k = \frac{n_k}{n}.$$

The slight difference is to estimate  $\mu_k$  and  $\Sigma$  by

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top.$$

# A plugin rule for estimating discriminant functions

- We use the plugin estimator

$$\hat{\delta}_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k, \quad \forall k \in C.$$

- The resulting LDA classifier is

$$\arg \max_{k \in C} \hat{\delta}_k(x).$$

# Logistic Regression v.s. LDA: similarity

For a binary classification problem, one can show that for LDA

$$\begin{aligned}\log\left(\frac{p_1(x)}{1 - p_1(x)}\right) &= \log\left(\frac{p_1(x)}{p_0(x)}\right) \\ &= c_0 + c_1x_1 + \cdots + c_px_p,\end{aligned}$$

also a linear form as logistic regression.

# Logistic Regression v.s. LDA: differences

1. LDA makes more assumption by specifying  $X \mid Y$ .
2. The parameters are estimated differently.
  - ▶ Logistic regression uses the conditional likelihood based on  $\mathbb{P}(Y|X)$  (known as discriminative learning).
  - ▶ LDA uses the full likelihood based on  $\mathbb{P}(X, Y)$  (known as generative learning).
3. If classes are well-separated, then logistic regression is not advocated.

# Other forms of Discriminant Analysis

LDA specifies

$$X \mid Y = k \sim N(\mu_k, \Sigma), \quad \forall k \in C.$$

Other discriminant analyses change the specifications for  $X \mid Y = k$ .

- **Quadratic discriminant analysis** (QDA) assumes

$$X \mid Y = k \sim N(\mu_k, \Sigma_k), \quad \forall k \in C,$$

by allowing different  $\Sigma_k$  across all classes.

- **Naive Bayes** assumes

$$X_1, \dots, X_p \text{ are independent given } Y = k.$$

For Gaussian density, this means that  $\Sigma_k$ 's are diagonal.

- Many other forms: different density models for  $X \mid Y = k$ , including non-parametric approaches.

# Quadratic Discriminant Analysis: $p = 1$

- Assume that

$$X \mid Y = k \sim N(\mu_k, \sigma_k^2), \quad \forall k \in C,$$

namely,

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2}.$$

- As a result,

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{\ell \in C} \pi_\ell f_\ell(x)} = \frac{\frac{\pi_k}{\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2}}{\sum_{\ell \in C} \frac{\pi_\ell}{\sigma_\ell} e^{-\frac{1}{2\sigma_\ell^2}(x-\mu_\ell)^2}}.$$

# Decision boundary of QDA

The Bayes rule classifies  $X = x$  to

$$\begin{aligned}\arg \max_{k \in C} p_k(x) &= \arg \max_{k \in C} \log(p_k(x)) \\&= \arg \max_{k \in C} \log \left[ \frac{\pi_k}{\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2} \right] \\&= \arg \max_{k \in C} \underbrace{-\frac{x^2}{2\sigma_k^2} + \frac{\mu_k}{\sigma_k^2}x - \frac{\mu_k^2}{2\sigma_k^2} + \log \pi_k - \log(\sigma_k)}_{\delta_k(x)}\end{aligned}$$

The name QDA is due to the fact that  $\delta_k(x)$  is **quadratic** in  $x$ .

# Quadratic Discriminant Analysis: $p \geq 1$

$$X \mid Y = k \sim N_p(\mu_k, \Sigma_k)$$

The discriminant function becomes

$$\begin{aligned}\delta_k(x) &= \log \left[ \frac{\pi_k}{|\Sigma_k|^{-1/2}} e^{-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)} \right] \\ &= x^\top \Sigma_k^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma_k^{-1} \mu_k + \log \pi_k - \frac{1}{2} x^\top \Sigma_k^{-1} x - \frac{1}{2} \log |\Sigma_k|.\end{aligned}$$

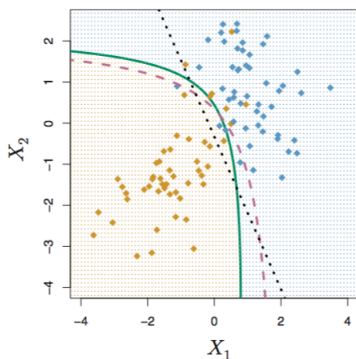
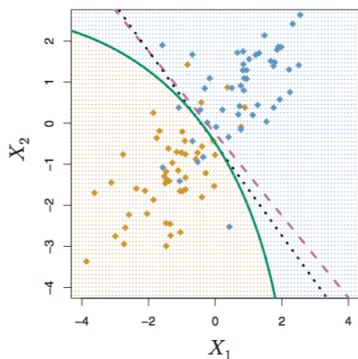
The **decision boundary** between any class  $k$  and class  $\ell$

$$\{x \in \mathbb{R}^p : \delta_k(x) = \delta_\ell(x)\}$$

is also quadratic in  $x$



# Decision boundaries of LDA and QDA



Decision boundaries of the Bayes classifier (purple dashed), LDA (black dotted), and QDA (green solid) in two scenarios.

# Estimation of QDA

Given training data  $(x_1, y_1), \dots, (x_n, y_n)$ , for any  $k \in C$ ,

- We have

$$n_k = \sum_{i=1}^n 1\{y_i = k\}.$$

- We estimate  $\pi_k$  by

$$\hat{\pi}_k = \frac{n_k}{n}.$$

- We estimate  $\mu_k$  and  $\Sigma$  by

$$\begin{aligned}\hat{\mu}_k &= \frac{1}{n_k} \sum_{i:y_i=k} x_i \\ \hat{\Sigma}_k &= \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top.\end{aligned}$$

- Plugin estimator for  $\delta(x)$ .

# Potential problems for LDA and QDA in high dimension

- LDA: we have

$$(K - 1) + pK + \frac{p(p - 1)}{2}$$

number of parameters to estimate.

- QDA: we have

$$(K - 1) + pK + \frac{p(p - 1)}{2}K$$

number of parameters to estimate.

- The estimation error is large when  $p$  is large comparing to  $n$ .

# Naive Bayes

**Naive Bayes** assumes that features are **independent** within each class, but not necessarily Gaussian.

- Useful when  $p$  is large, whence QDA and even LDA break down.
- Under Gaussian distributions, naive Bayes assumes

$$\Sigma_k = \text{diag}(\sigma_{k1}^2, \dots, \sigma_{kp}^2), \quad \forall k \in C.$$

The discriminant function is

$$\delta_k(x) = -\frac{1}{2} \sum_{j=1}^p \frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log \pi_k.$$

- It is easy to deal with both quantitative and categorical features.
- Despite the strong independence assumption within class, naive Bayes often produces good classification results.