STAT406- Methods of Statistical Learning
Lecture 12

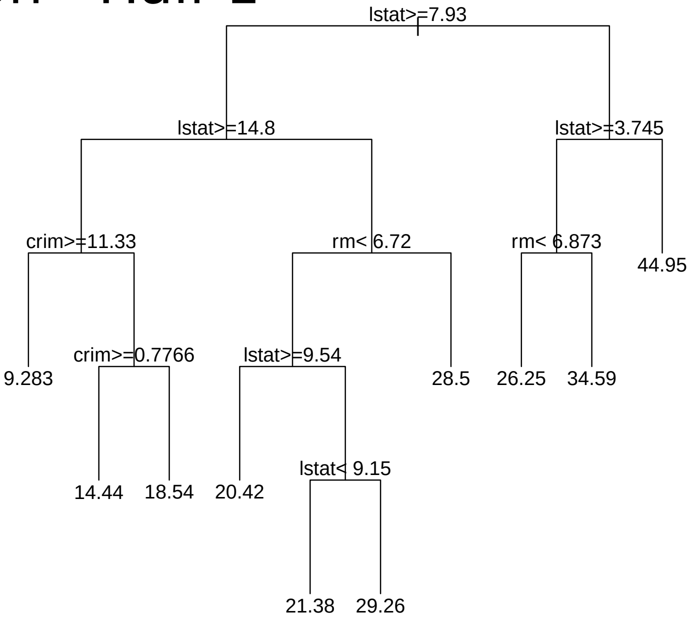
Matias Salibian-Barrera

UBC - Sep / Dec 2017

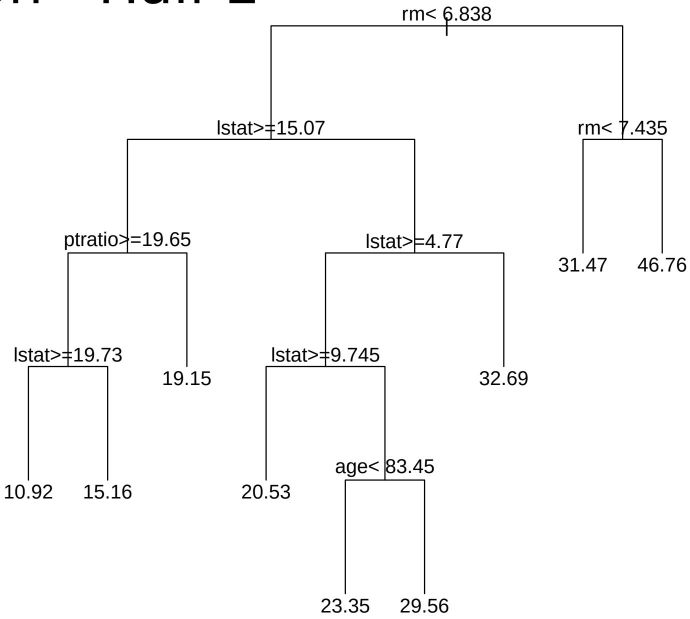
Bagging

- Trees can be highly variable
- Trees computed on samples from the sample population can be quite different from each other
- For example, we split the Boston data in two...

Boston - Half 1



Boston - Half 2



Bagging

- Linear regression, for example, is not so variable
- Estimated coefficients computed on the same two halfs

```
     (Intercept)  crim   zn indus chas
[1,]       39.21 -0.13 0.04  0.04 2.72
[2,]       33.12 -0.10 0.05 -0.01 2.80
              nox   rm age   dis  rad   tax
[1,]       -20.07 3.45   0 -1.44 0.28 -0.01
[2,]       -14.18 4.15   0 -1.46 0.34 -0.02
        ptratio black lstat
[1,]   -1.01  0.01 -0.56
[2,]   -0.90  0.01 -0.50
```


Bagging

- If we could average many trees trained on independent samples from the same population, we would obtain a predictor with lower variance

- If $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_B$ are B regression trees, then their average is

$$\hat{f}_{\text{av}}(\mathbf{x}) = \frac{1}{B} \sum_{j=1}^{B} \hat{f}_j(\mathbf{x})$$

Bagging

- However, we generally do not have B training sets...
- We can **bootstrap** the training set to obtain B pseudo-new-training sets
- Let $(Y_1, \mathbf{X}_1)$, $(Y_2, \mathbf{X}_2)$, $\ldots$, $(Y_n, \mathbf{X}_n)$ be the training sample, where

$$(Y_j, \mathbf{X}_j) \ \sim \ F_0$$

Bagging

- If we knew F_0, then we could generate / simulate new training sets, and average the resulting trees...
- We do not know F_0, but we have an estimate for it
- Let F_n be the empirical distribution of our only training set $(Y_1, \mathbf{X}_1)$, $(Y_2, \mathbf{X}_2)$, $\ldots$, $(Y_n, \mathbf{X}_n)$

Bagging

- We know that

$$F_n \xrightarrow[n \to \infty]{} F_0$$

(in what sense?)

- Bootstrap generates / simulates samples from F_n
- Taking a sample of size n from F_n is the same as sampling with replacement from the training set $(Y_1, \mathbf{X}_1)$, $(Y_2, \mathbf{X}_2)$, $\ldots$, $(Y_n, \mathbf{X}_n)$

Bagging

- To apply bagging to a regression tree, take B independent samples (with replacement) from the training set
- Obtain the B trees: $\hat{f}_1^*, \hat{f}_2^*, \ldots, \hat{f}_B^*$
- and average their predictions

$$\hat{f}_{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_{j=1}^{B} \hat{f}_j^*(\mathbf{x})$$

Bagging

- Generally, we apply bagging on "large" trees, without pruning them (try to retain their low-bias and reduce their variance by averaging)
- With the Boston data set, if we apply bagging to the regression tree computed on the training set, and then use it to predict on the test set, we obtain:

Bagging

- $B = 1$
```
> mean((dat.te$medv - pr.ba)^2)
[1] 16.44972
```
- $B = 5$
```
> mean((dat.te$medv - pr.ba)^2)
[1] 15.12332
```
- $B = 100$
```
> mean((dat.te$medv - pr.ba)^2)
[1] 12.30543
```
- $B = 500$
```
> mean((dat.te$medv - pr.ba)^2)
[1] 12.32504
```

Bagging

- $B = 2000$

```
> mean((dat.te$medv - pr.ba)^2)
[1] 11.8116
```

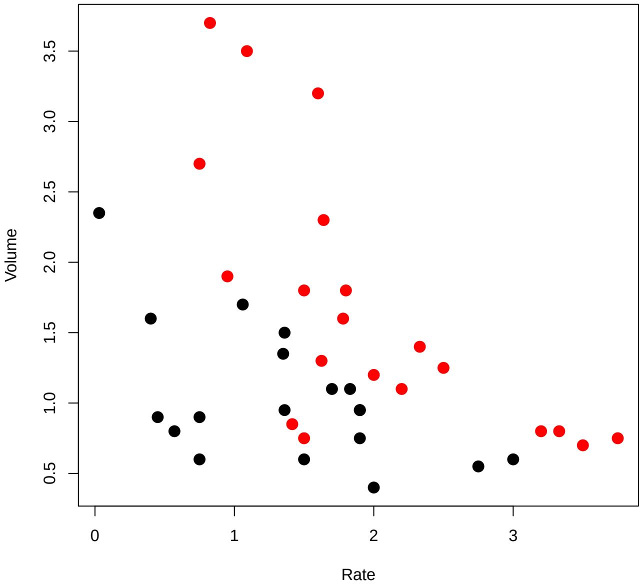
- $B = 5000$

```
> mean((dat.te$medv - pr.ba)^2)
[1] 11.85943
```

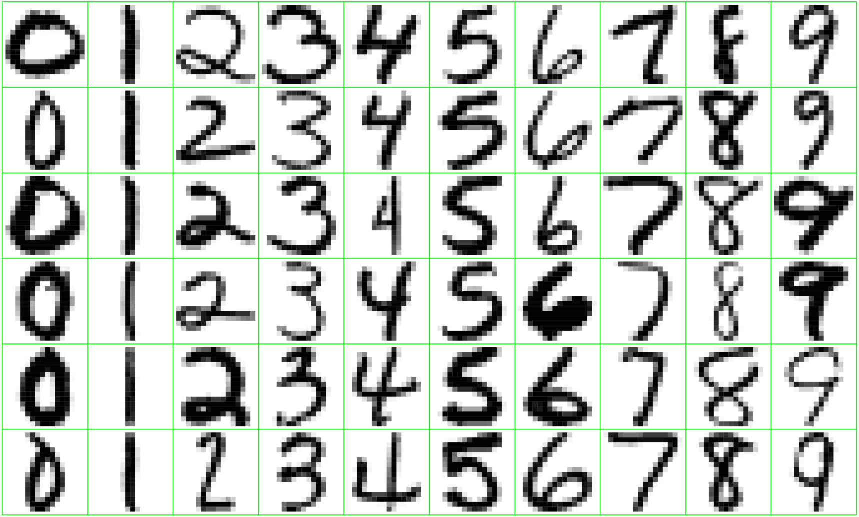

Bagging

- This approach applies to any predictor (not only trees)
- It will be particularly useful for low-bias / high-variance predictors

Classification



Predict hand-written digits



Classification as prediction

- In general, we have n observations (training)
- $(g_1, \mathbf{x}_1)$, $(g_2, \mathbf{x}_2)$, $\ldots$, $(g_n, \mathbf{x}_n)$
- we would like to build a classifier, a function $\hat{g}(\mathbf{x})$ to predict the true class g of a future observation $(g, \mathbf{x})$ (for which g is unknown)

Classification as prediction

- In general, there are K possible classes, c_1, c_2, $\ldots$, c_K. In other words $g \in \{c_1, c_2, \ldots, c_K\}$
- Consider the following loss function

$$L(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{if } a \neq b \end{cases}$$

Classification as prediction

- Find a classifier $\hat{g}(\mathbf{x})$ such that

$$E_{(G,\mathbf{X})}\left[L\left(G, \hat{g}(\mathbf{x})\right)\right] \; \leq \; E_{(G,\mathbf{X})}\left[L\left(G, h(\mathbf{X})\right)\right]$$

for any other function h

$$\begin{aligned} E_{(G,\mathbf{X})}\left[L\left(G, \hat{g}(\mathbf{X})\right)\right] &= E_{\mathbf{X}}\left\{E_{G|\mathbf{X}}\left[L\left(G, \hat{g}(\mathbf{X})\right)\right]\right\} \\ &= E_{\mathbf{X}}\left\{\sum_{j=1}^{K} L\left(c_j, \hat{g}(\mathbf{X})\right) P\left(G = c_j \middle| \mathbf{X}\right)\right\} \end{aligned}$$

Classification as prediction

- It is sufficient to find $\hat{g}(\mathbf{X})$ that minimizes

$$\sum_{j=1}^{K} L\left(c_j, \hat{g}(\mathbf{X})\right) P\left(G = c_j \middle| \mathbf{X}\right)$$

$$= \sum_{c_j \neq \hat{g}(\mathbf{X})} P\left(G = c_j \middle| \mathbf{X}\right)$$

$$= 1 - P\left(G = \hat{g}(\mathbf{X}) \middle| \mathbf{X}\right)$$

- Hence, the optimal classifier satisfies

$$P\left(G = \hat{g}(\mathbf{X}) \middle| \mathbf{X}\right) \geq P\left(G = c_j \middle| \mathbf{X}\right) \quad \text{for all } c_j$$

More than 2 groups

- In other words, $\hat{g}(\mathbf{X})$ should be the class with the highest probability

$$\hat{g}(\mathbf{X}) \; = \; \arg \max_{\mathbf{g} \in \{c_1, \ldots, c_K\}} P\Big(G = \mathbf{g} \,|\, \mathbf{X} \Big)$$

- "Assign $\mathbf{X}$ to the class with largest posterior probability given $\mathbf{X}$"

Classification as prediction

- Most classifiers can be thought of as different ways to estimate or model

$$\mathbf{f}_{\mathbf{j}}(\mathbf{x}) = P\Big(G = \mathbf{c}_j \Big| \mathbf{X} = \mathbf{x}\Big)$$

- For example, logistic classifiers propose a model for $\mathbf{f}_j$:

$$\mathbf{f}_{\mathbf{j}}(\mathbf{x}) = \frac{\exp\left(\boldsymbol{\beta}_{\mathbf{j}}\,\mathbf{x}\right)}{1 + \exp\left(\boldsymbol{\beta}_{\mathbf{j}}\,\mathbf{x}\right)}$$

Classification as prediction

- Vaso example - Logistic linear model
- Data $(y_1, \mathbf{x}_1)$, $(y_2, \mathbf{x}_2)$, ..., $(y_n, \mathbf{x}_n)$
- $y_j = 0, 1$, $\mathbf{x} = (\text{rate}, \text{volume})'$
- A possible model is

$$P\left(y_j = 1 \middle| \mathbf{x}_j\right) = \frac{\exp\left(\beta' \mathbf{x}_j\right)}{1 + \exp\left(\beta' \mathbf{x}_j\right)}$$

Classification as prediction

- We can estimate β using MLE
- Function `glm` in `R`
- Given values of `rate` and `volume` we predict a 1 if

$$\hat{P}\left(y_j = 1 \middle| \text{rate}, \text{volume}\right) \ > \ 0.5$$

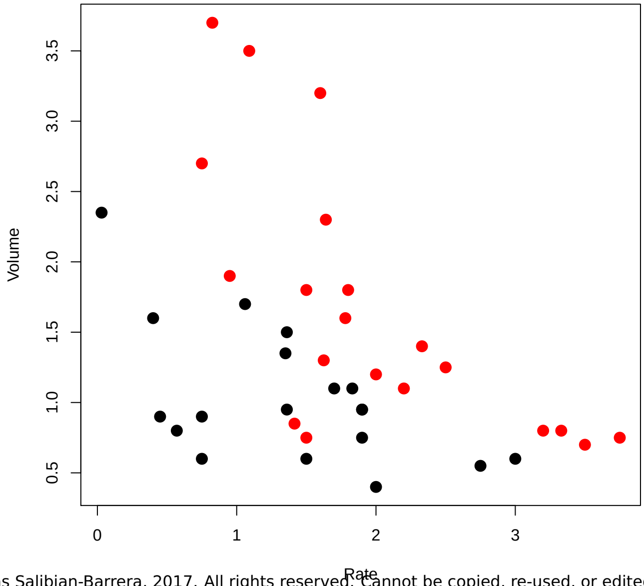
Classification as prediction

- These posterior probabilities

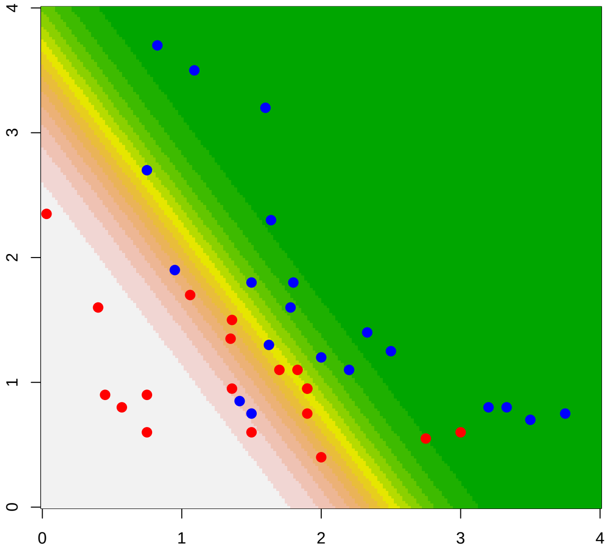
$$P\left(G = \mathbf{c}_j \,\middle|\, \mathbf{X} = \mathbf{x}\right)$$

  can also be used
- to quantify uncertainty in the classification for a particular value of $\mathbf{x}$
- to identify regions of the feature space where classification isn't so clear

Example - Vaso data



Logistic based probabilities



A model for $\mathbf{X}|g$

If we **model** the feature **distribution** in each **group**:

$$f(\mathbf{X}|\, G = c_{\mathbf{k}}) = f_{\mathbf{k}}(\mathbf{X}) \qquad \mathbf{k} = 1, \ldots, \mathbf{K}$$

then

$$P(G = c_{\mathbf{k}}|\, \mathbf{X}) = \frac{f(\mathbf{X}|\, G = c_{\mathbf{k}})\, p_{\mathbf{k}}}{f(\mathbf{X})} = \frac{f_{\mathbf{k}}(\mathbf{X})\, p_{\mathbf{k}}}{f(\mathbf{X})}$$

thus

$$\hat{\mathbf{g}}(\mathbf{X}) = \arg \max_{1 \leq \mathbf{k} \leq \mathbf{K}} f_{\mathbf{k}}(\mathbf{X})\, p_{\mathbf{k}}$$

A model for $\mathbf{X}|g$

For example, we can assume that

$$\mathbf{X} | G = c_{\mathbf{k}} \sim \mathcal{N}(\mu_{\mathbf{k}}, \mathbf{\Sigma})$$

then, we can estimate

$$\hat{f}_{\mathbf{k}}(\mathbf{X}) \sim \mathcal{N}\left(\hat{\mu}_{\mathbf{k}}, \widehat{\mathbf{\Sigma}}\right)$$

using the sample mean of each group and the pooled sample covariance matrix.

We can then find the class $\mathbf{k}$ that has the largest $\hat{f}_{\mathbf{k}}(\mathbf{X})\, p_{\mathbf{k}}$

Gaussian populations

Note that if $f_j \sim \mathcal{N}_p\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}\right)$, $j = 1, 2$

$$f_1(\mathbf{x})\, p_1 \; > \; f_2(\mathbf{x})\, p_2 \quad \Leftrightarrow$$

$$\log\left(\frac{f_1(\mathbf{x})\, p_1}{f_2(\mathbf{x})\, p_2}\right) > 0 \quad \Leftrightarrow$$

$$\mathbf{a}'\mathbf{x} + b \; > \; 0$$

for some $\mathbf{a} \in \mathbb{R}^p$ and $b \in \mathbb{R}$.

In other words, boundaries between classes are **linear**.

Gaussian populations

Furthermore, we can estimate this linear boundary because

$$\mathbf{a} = \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right)$$

and

$$b = -\frac{1}{2} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right)' \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 \right) - \log \left(\frac{p_2}{p_1} \right)$$

Gaussian populations

We can also write this in term of class probabilities

$$\frac{P\left(G = c_1 \middle| \mathbf{X}\right)}{P\left(G = c_2 \middle| \mathbf{X}\right)} > 1 \quad \Leftrightarrow \quad f_1(\mathbf{x})\, p_1 \; > \; f_2(\mathbf{x})\, p_2$$

$$\Leftrightarrow \quad \log\left(\frac{f_1(\mathbf{x})\, p_1}{f_2(\mathbf{x})\, p_2}\right) > 0 \quad \Leftrightarrow \quad \mathbf{a}'\mathbf{x} + b \; > \; 0$$

Gaussian populations

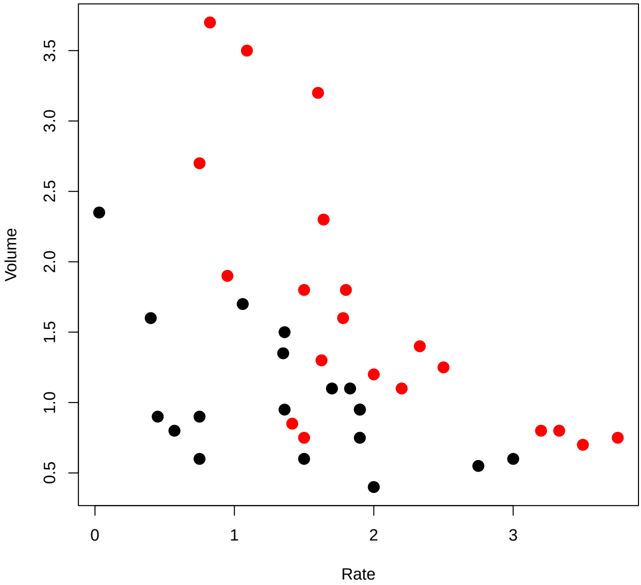
In fact, for normally distributed features we have

$$\log\left(\frac{P\left(G=c_1|\,\mathbf{X}\right)}{P\left(G=c_2|\,\mathbf{X}\right)}\right) =$$

$$\log\left(\frac{P\left(G=c_1|\,\mathbf{X}\right)}{1-P\left(G=c_1|\,\mathbf{X}\right)}\right) = \mathbf{a}'\mathbf{x}+b$$

With two classes, we have also estimated **a** and b using logistic regression.

Example - Vaso data



Example - Vaso data

- First assume that `Volume` and `Rate` are normally distributed in each class
- Then, the optimal classifier classifies a point $\mathbf{x} = (\text{Volume}, \text{Rate})'$ in class 1 (red) if

$$\mathbf{a}'\mathbf{x} + b \; > \; 0$$

where

$$\mathbf{a} = \mathbf{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right)$$

and

$$b = -\frac{1}{2} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right)' \, \mathbf{\Sigma}^{-1} \left(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 \right) - \log \left(\frac{p_2}{p_1} \right)$$

Example - Vaso data

- We can estimate $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}$ (and even p_1 and p_2). **How?**
- We get $\hat{\mathbf{a}} = (-2.77, -2.37)'$ and $\hat{b} = 7.72$
- Then, the estimated optimal classifier classifies a point $\mathbf{x} = (\text{Volume}, \text{Rate})'$ in class 1 (red) if

$$-2.77\,\text{Volume} - 2.37\,\text{Rate} + 7.72 > 0$$

Example - Vaso data

- Furthermore

$$\widehat{P}(G = 1 | (\text{Volume}, \text{Rate})) = \frac{\exp(-2.77\,\text{Volume} - 2.37\,\text{Rate} + 7.72)}{1 + \exp(-2.77\,\text{Volume} - 2.37\,\text{Rate} + 7.72)}$$

and

$$\widehat{P}(G = 2 | (\text{Volume}, \text{Rate})) = 1 - \widehat{P}(G = 1 | (\text{Volume}, \text{Rate})) =$$

Example - Vaso data

- Now, create a fine grid of `Volume` and `Rate` values, and use the previous formulas to predict

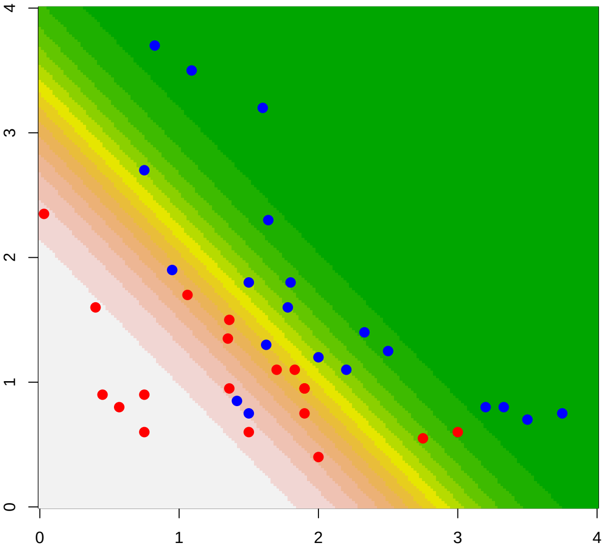
$$P\left(G = j \,|\, (\text{Volume}, \text{Rate})\right) \; , \quad j = 1, 2$$

- Plot these posterior probabilities
- We can do this by hand, or using the function `lda` in package `MASS` and its `predict` method

Example - Vaso data

```
library(MASS)

data(vaso, package='robustbase')

plot(Volume ~ Rate, pch=19, col=c('red', 'blue')[Y+1],
      data=vaso, cex=1.3)

a.lda <- lda(Y ~ Volume + Rate, prior = c(.5, .5),
     data=vaso)

aa <- seq(0, 4, length=200)
bb <- seq(0, 4, length=200)
dd <- expand.grid(aa, bb)
names(dd) <- c('Volume', 'Rate')

pr.lda <- predict(a.lda, newdata=dd)$posterior[,1]
image(aa, bb, matrix(pr.lda, 200, 200),
     col=terrain.colors(15), xlab='', ylab='')
points(Volume ~ Rate, pch=19, col=c('red', 'blue')[Y+1],
     data=vaso, cex=1.3)
```

Gaussian-based probabilities



Example - Vaso data

- Note that if we do not assume Gaussian features but insist that

$$\log\left(\frac{P\left(G=1|\mathbf{X}\right)}{P\left(G=2|\mathbf{X}\right)}\right) = \log\left(\frac{P\left(G=1|\mathbf{X}\right)}{1-P\left(G=1|\mathbf{X}\right)}\right) = \mathbf{a}'\mathbf{x}+b$$

we can use `glm` to estimate $\hat{\mathbf{a}}$ and $\hat{b}$:

$\hat{\mathbf{a}} = (-3.88, -2.65)'$ and $\hat{b} = 9.53$

Logistic-based probabilities

```
data(vaso, package='robustbase')

a <- glm(Y ~ Volume + Rate, data=vaso, family=binomial)

aa <- seq(0, 4, length=200)
bb <- seq(0, 4, length=200)
dd <- expand.grid(aa, bb)
names(dd) <- c('Volume', 'Rate')

yy <- predict(a, newdata=dd, type='response')

image(aa, bb, matrix(1-yy, 200, 200),
     col=terrain.colors(15), xlab='', ylab='')
points(Volume ~ Rate, pch=19, col=c('red', 'blue')[Y+1]
     data=vaso, cex=1.3)
```

Logistic-based probabilities