# Supervised Learning: Classification

Noah Simon & Ali Shojaie

July 20-22, 2016
Summer Institute for Statistics of Big Data
University of Washington

# Classification

- Regression involves predicting a continuous-valued response.
- Classification involves predicting a categorical / qualitative response:
  - Cancer versus Normal
  - Tumor Type 1 versus Tumor Type 2 versus Tumor Type 3
- Classification problems tend to occur even more frequently than regression problems in biomedical applications.
- Just like regression,
  - Classification cannot be blindly performed in high-dimensions because you will get zero training error but awful test error;
  - Properly estimating the test error is crucial; and
  - There are a few tricks to extend classical classification approaches to high-dimensions, which we have already seen in the regression context!

# Classification

- Categorical / qualitative variables take values in an unordered set: e.g.
  eye color $\in \{$*brown*, *blue*, *green*$\}$
  email $\in \{$*spam*, *not spam*$\}$.
- We want to build a function that takes as input the feature vector $X$ and predicts the value for $Y$.
- Often we are more interested in estimating the probability that $X$ belongs to a given category.
- For example: we might want to know the probability that someone will develop diabetes, rather than to predict whether or not they will develop diabetes.

# Can't We Just Use Linear Regression?

- Classify an emergency room patient on the basis of her symptoms to one of three conditions:

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

- If we apply linear regression, then the results will depend on the choice of coding . . . and the coding implies an ordering among the medical conditions.
- A classification approach is more appropriate.

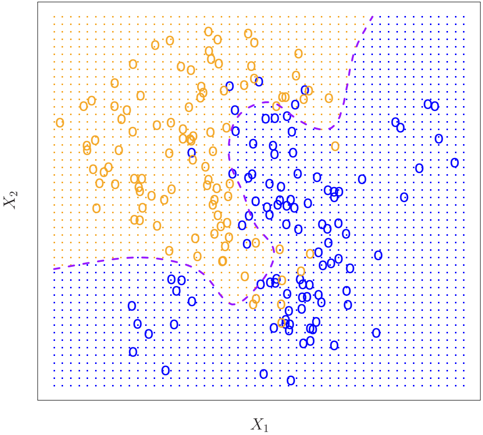
# Classification

- There are many approaches out there for performing classification.
- We will discuss three: k-nearest neighbors, logistic regression, and support vector machines.

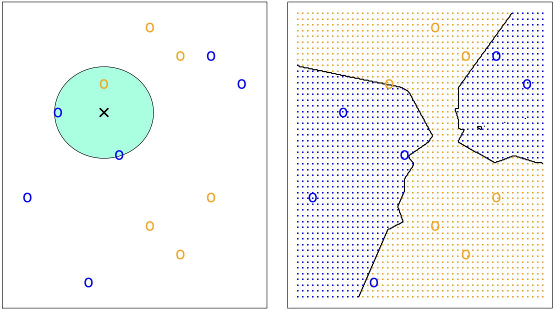
# $K$-Nearest Neighbors

- Can I take a totally non-parametric (model-free) approach to classification?
- K-nearest neighbors:
  1. Identify the $K$ observations whose $X$ values are closest to the observation at which we want to make a prediction.
  2. Classify the observation of interest to the most frequent class label of those $K$ nearest neighbors.

# *K*-Nearest Neighbors

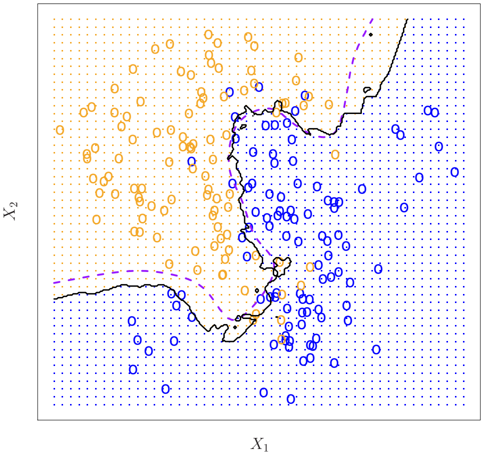


# *K*-Nearest Neighbors

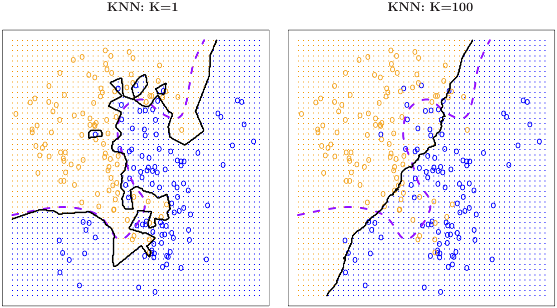


# *K*-Nearest Neighbors

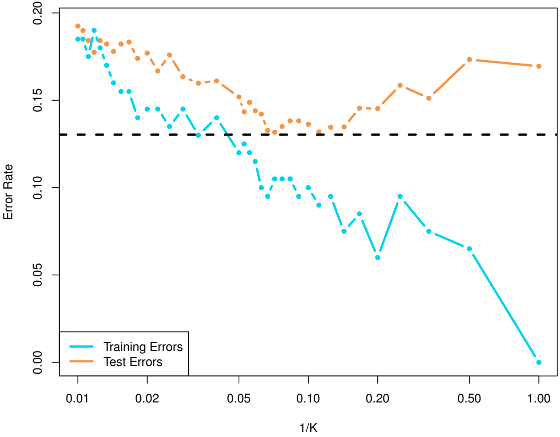
**KNN: K=10**



# *K*-Nearest Neighbors



# $K$-Nearest Neighbors



# $K$-Nearest Neighbors

- Simple, intuitive, model-free.
- Good option when $p$ is very small.
- Curse of dimensionality: when $p$ is large, no neighbors are "near". All observations are close to the boundary.
- Do not use in high dimensions!

# Logistic Regression

- Logistic regression is the straightforward extension of linear regression to the classification setting.
- For simplicity, suppose $y \in \{0, 1\}$: a two-class classification problem.
- The simple linear model $y = X\beta + \epsilon$ doesn't make sense for classification.

# Logistic Regression

- Let $p(X) = \Pr(Y = 1|X)$.
- Suppose we want to use biomarker level to predict probability of cancer.
- Logistic regression uses the form

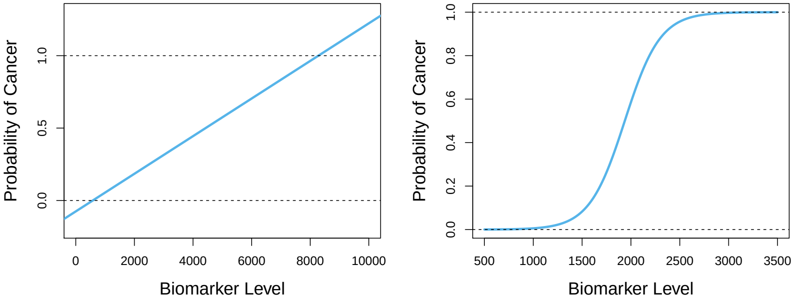
$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

- $p(X)$ will lie between 0 and 1.
- Furthermore,

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

- This function of $p(X)$ is called the logit or log odds.

# Why Not Linear Regression?



- Left: linear regression.
- Right: logistic regression.

# Multiple Logistic Regression

- Just like before:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

- And just like before:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

# Example in R

```
xtr <- matrix(rnorm(1000*20),ncol=20)
beta <- c(rep(1,10),rep(0,10))
ytr <- 1*((xtr%*%beta + .2*rnorm(1000)) >= 0)
mod <- glm(ytr˜xtr,family="binomial")
print(summary(mod))
```

# Five Ways to Extend Logistic to High Dimensions

1. Variable Pre-Selection
2. Forward Stepwise Logistic Regression
3. Ridge Logistic Regression
4. Lasso Logistic Regression
5. Principal Components Logistic Regression

How to decide which approach is best, and which tuning parameter value to use for each approach? Cross-validation or validation set approach.

# What is an appropriate validation measure?

For classification without a probability or score:

- Misclassification rate:

$$\frac{\text{\#test samples misclassified}}{\text{total \# of test samples}}$$

# What is an appropriate validation measure?

For probablistic classification

- Can still use misclassification rate.
- Like in continuous regression could use SSE:

$$\sum_{i\in\mathrm{test}} \left(y_i - \hat{p}_i\right)^2$$

- Often preferable to use "predictive [log]likelihood":

$$-\log\left[\prod_{i\in\mathrm{test}} \hat{p}_i^{y_i}\left(1-\hat{p}_i\right)^{1-y_i}\right]$$

- Can also use ROC-curve-based metric (eg. AUC)

Remember though; all of these must be conducted on a separate validation set.

# Example in R: Lasso Logistic Regression

```
xtr <- matrix(rnorm(1000*20),ncol=20)
beta <- c(rep(1,5),rep(0,15))
ytr <- 1*((xtr%*%beta + .5*rnorm(1000)) >= 0)
cv.out <- cv.glmnet(xtr, ytr, family="binomial", alpha=1)
plot(cv.out)
```

## Let's Try It Out in R!

Chapter 4 R Lab
Skip part on LDA & QDA
www.statlearning.com

# Bayes-based classifiers

Suppose rather than knowing $P\left(y=j|x\right)$...

we have information on $f_j(x) = P\left(x|y=j\right)$, the feature distribution within each class

How do we use this to make predictions?

Using Bayes Theorem:

$$P\left(y=j|x\right) = \frac{f_j(x)\pi_j}{\sum_k f_k(x)\pi_k}$$

here $\pi_k = P(y=k)$ is the prior probability of class $k$.

# Estimating the Rule

To apply Bayes Theorem

$$P\left(y = j|x\right) = \frac{f_j(x)\pi_j}{\sum_k f_k(x)\pi_k}$$

we need

- $f_k(x)$ for $k = 1, \ldots, K$
- $\pi_k$ for $k = 1, \ldots, K$

# Estimating the $\pi_k$

$\pi_k$ is generally simple to estimate

- If your data are a random sample; then can use the sample proportion
$$\hat{\pi}_k = \frac{\#\left\{y_i = k\right\}}{n}$$
- Otherwise can use outside information (eg. historical data)

If you change population proportions; it is easy to adjust the rule.

# Estimating the $f_k(x)$

Estimate of $f_k(x) = P\left(x|y=k\right)$ is more difficult.

This is a density estimation problem.

The tools we discuss for this break down into 3 general categories

- flexible, non-parametric estimates
- parametric estimates
- shrunken parametric estimates

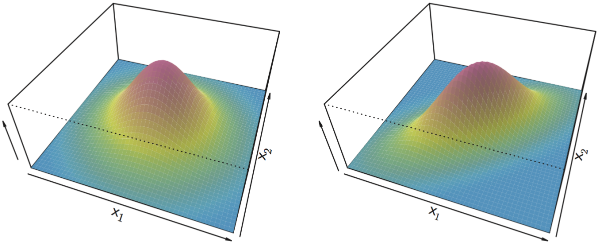
The above are ordered (more-or-less) by where they fall on bias/variance spectrum:

more flexible → less bias/more variance

# Parametric $f_k(x)$ Estimate

Most well known estimator of this type is Linear/Quadratic Discriminant Analysis:

Here we assume that $f_k(x)$ is Gaussian density, $N(\mu_k, \Sigma_k)$



# Discriminant Analysis

There are three main types of unpenalized discriminant analysis:

- Quadratic (QDA)
- Linear (LDA)
- Diagonal (DDA)

These make different assumptions on the covariance structure:

- QDA makes no assumptions
- LDA assumes a pooled variance $\Sigma = \Sigma_k$ for all $k$
- DDA assumes a pooled variance; and further that $\Sigma$ is diagonal (i.e. no correlation among covariates!)

# Discriminant Analysis

Why would we choose DDA over QDA?

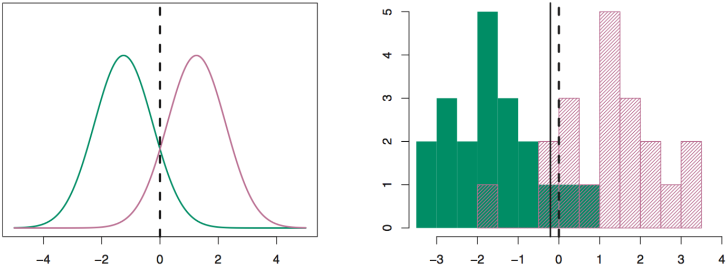
Remember, *flexibility* comes at a price!

QDA will have the least bias; but has many more parameters to estimate

Often good estimates of the correlation don't improve classifications much

DDA takes into account the scale of each feature, but trades a bit of bias for potentially a large reduction in variance

# LDA for $p = 1$

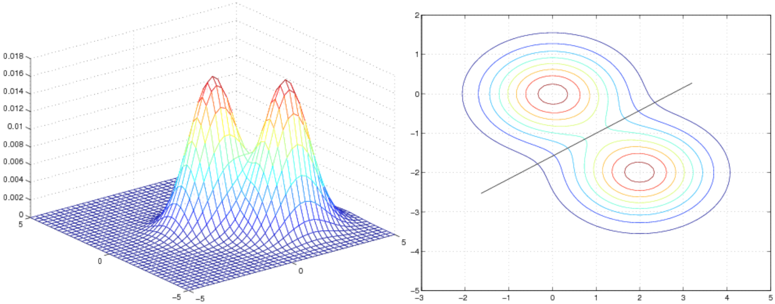


- To make this work, we need to estimate the parameters. The ML estimates are given by $\hat{\pi}_k = n_k/n$ and

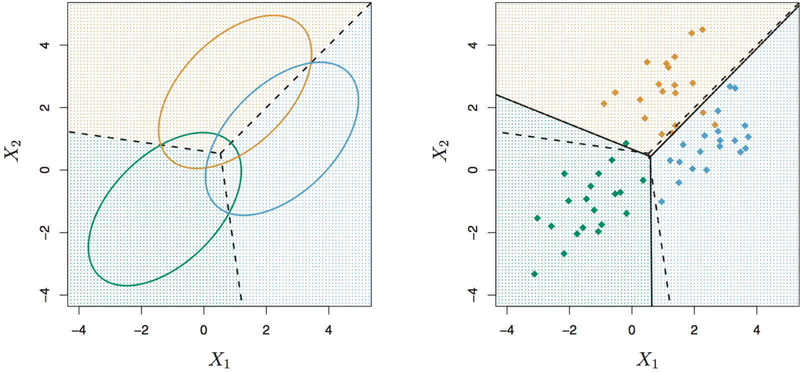
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \qquad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

- The picture is very similar if $K > 2$...or if $p > 1$

# LDA for $p > 1$


0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
5
0
−5
−5
0
5
2
1
0
−1
−2
−3
−4
−5
−3
−2
−1
0
1
2
3
4
5

# LDA for $p > 1$

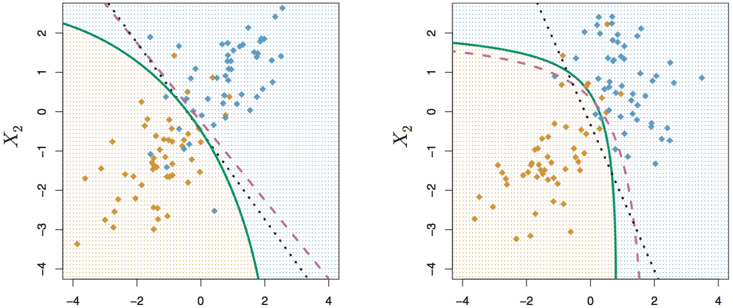


# QDA vs LDA

The level-curves for each class look identical with LDA;

QDA allows for different classes to have differently shaped ellipsoids...

This results in decision boundaries that are non-linear (quadratic in fact)



# DDA

For DDA...

- level curves are spheres (not ellipsoids).
- decision boundaries are still linear
- sometimes called *naive bayes* (that doesn't mean it's bad though!)
- with $\pi_k = \frac{1}{K}$ for all $k$, and equal variances (ie. $\Sigma = \sigma I$); this is just the *nearest centroid* classifier

# -DA vs logistic regression

Discriminant Analysis model can actually be rewritten as multinomial logistic models:

Beginning with

$$P\left(y=j|x\right)=\frac{f_j(x)\pi_j}{\sum_k f_k(x)\pi_k}$$

and

$$f_k(x)\propto\exp\left[-\frac{1}{2}\left(x-\mu_k\right)^\top\Sigma_k^{-1}\left(x-\mu_k\right)\right]$$

substituting and simplifying we get

$$P\left(y=j|x\right)=\frac{e^{\eta_j}}{\sum_k e^{\eta_k}}$$

# -DA vs logistic regression

$$P\left(y = j|x\right) = \frac{e^{\eta_j}}{\sum_k e^{\eta_k}}$$

where

$$\eta_k = \beta_0 + x^\top \beta + x^\top \Sigma_k^{-1} x$$

This is just a multinomial logistic model with quadratic terms and interactions.

In particular for LDA (where $\Sigma_k = \Sigma$ is pooled) we have cancellation and get

$$\eta_k = \beta_0 + x^\top \beta$$

Simply a linear logistic model.

# Shrunken Parametric Estimates

Sometimes the optimal bias/variance tradeoff is between two parametric classes.

For example: We may not have the data to estimate completely different covariance matrices for each class (i.e. QDA); but we may not want to use identical covariance matrices.

In this case we can take a weighted combination of our estimates. This is called regularized discriminant analysis.

This is a type of *shrunken parametric estimator*.

# Regularized Discriminant Analysis

For shrinking between QDA/LDA we use:

$$\hat{\Sigma}_k^{RDA} = \lambda \hat{\Sigma}^{LDA} + (1 - \lambda)\hat{\Sigma}_k^{QDA}$$

For shrinking between LDA and Naive Bayes we use

$$\hat{\Sigma}^{RDA} = \lambda \hat{\Sigma}^{LDA} + (1 - \lambda)\hat{\Sigma}^{NB}$$

$\lambda$ is a tuning parameter, and is generally selected via CV

# DA in High Dimensions

All of the Discriminant Analysis techniques discussed so far use all the features.

For high dimensional problems this will lead to over-fitting

One popular solution is to shrink each class-mean estimate $\hat{\mu}_k$ towards the overall mean $\hat{\mu}$ using element-wise soft-thresholding

This method is called Nearest Shrunken Centroids (though it should probably more appropriately be "nearest shrunken DDA")

# Nearest Shrunken Centroids (PAM)

Steps to the method:

1. Calculate our pooled, diagonal estimate of $\Sigma$; let $s_j$ be the sd. estimate of gene $j$
2. Calculate the within class mean $\hat{\mu}_{jk}$ for each gene $j$, class $k$, and overall mean $\hat{\mu}_{j\cdot}$
3. Set $\hat{\mu}_{jk}^{PAM}$ to be the shrunken difference:

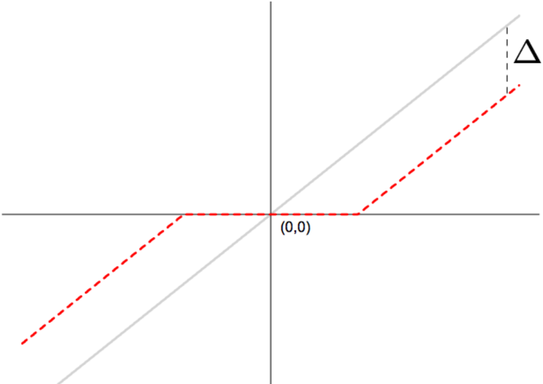
$$\hat{\mu}_{jk}^{PAM} = \hat{\mu}_{j\cdot} + s_j * SHRINK_{\Delta}\left(\frac{\hat{\mu}_{jk} - \hat{\mu}_{j\cdot}}{s_j}\right)$$

where $SHRINK_{\Delta}$ is the *Soft Thresholding Function*

## Soft Thresholding

The soft thresholding shrinks its argument towards 0 — if it hits 0; then it stops!

Can be thought of as the continuous version of usual *thresholding*



# Other Regularized DA Methods

- Recall that in PAM, $\hat{\mu}_{jk}$'s are soft thresholded towards the common mean $\hat{\mu}_{j.}$
- Alternatively, $\mu_{jk}$'s can be penalized
  - zero, using a lasso penalty

$$\sum_{j} \sum_{k} |\mu_{jk}|,$$

  - or towards each other, using a fused lasso penalty

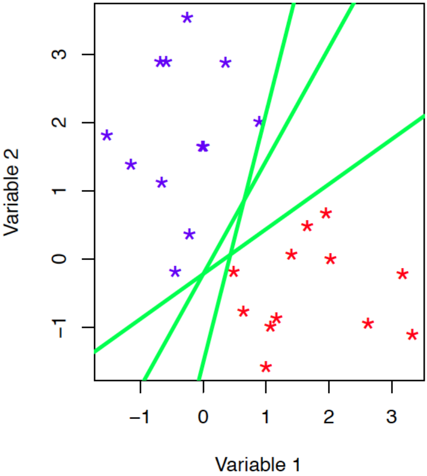
$$\sum_{j} \sum_{k,k'} |\mu_{jk} - \mu_{jk'}|.$$

  Both of these are implemented in R-package `penalizedLDA`.
- Another option, which is especially helpful when using QDA is to penalize the covariance matrices $\Sigma_k$ (or their inverses).

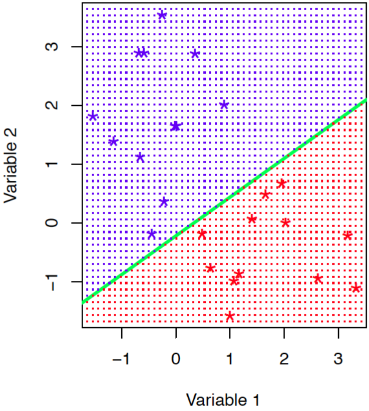
# Support Vector Machines

- Developed in around 1995.
- Touted as “overcoming the curse of dimensionality.”
- Does not automatically overcome the curse of dimensionality!!!
- Fundamentally and numerically very similar to logistic regression.
- But, it is a nice idea.

# Separating Hyperplane

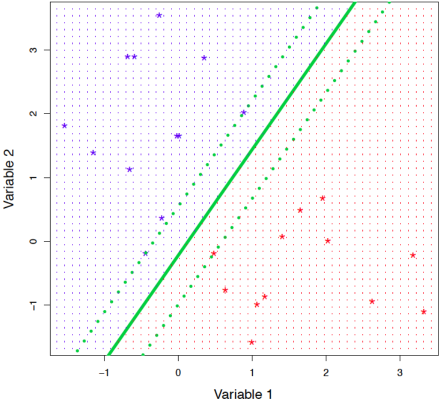


# Classification Via a Separating Hyperplane



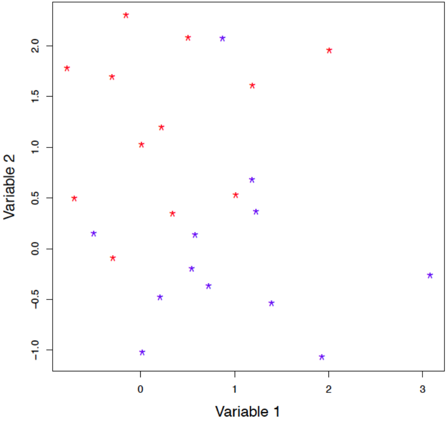
Blue class if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > c$; red class otherwise.

# Maximal Separating Hyperplane

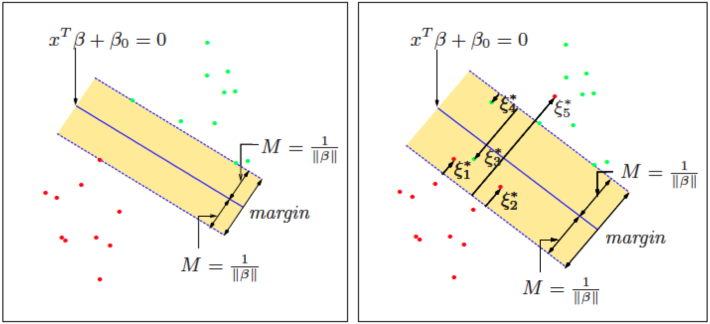


Note that only a few observations are on the margin: these are the support vectors.

# What if There is No Separating Hyperplane?



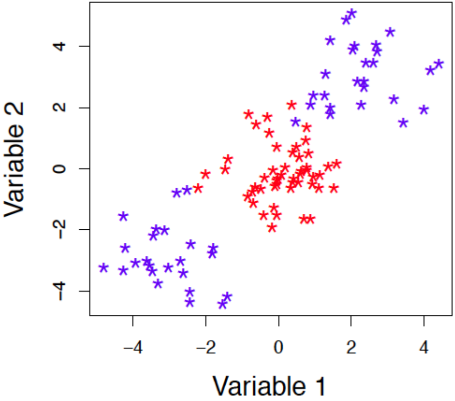
# Support Vector Classifier: Allow for Violations



# Support Vector Machine

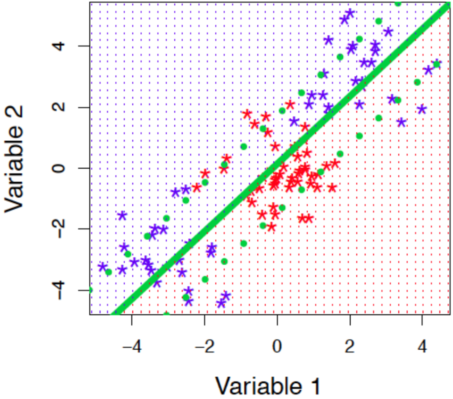
- The support vector machine is just like the support vector classifier, but it elegantly allows for non-linear expansions of the variables: “non-linear kernels”.
- However, linear regression, logistic regression, and other classical statistical approaches can also be applied to non-linear functions of the variables.
- For historical reasons, SVMs are more frequently used with non-linear expansions as compared to other statistical approaches.

# Non-Linear Class Structure



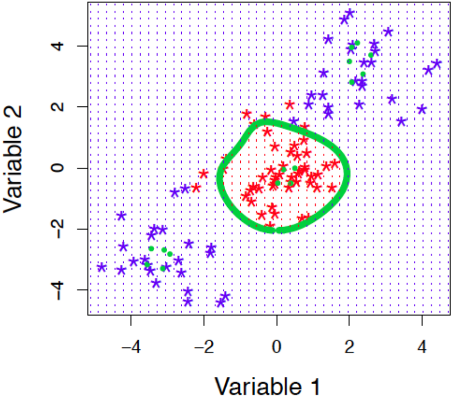
This will be hard for a linear classifier!

# Try a Support Vector Classifier



Uh-oh!!

# Support Vector Machine

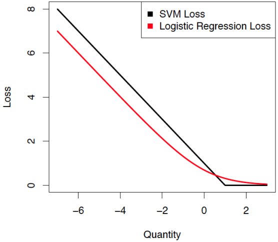


Much Better.

# Is A Non-Linear Kernel Better?

- Yes, if the true decision boundary between the classes is non-linear, and you have enough observations (relative to the number of features) to accurately estimate the decision boundary.
- No, if you are in a very high-dimensional setting such that estimating a non-linear decision boundary is hopeless.

# Support Vector Classifier Versus Logistic Regression



- Bottom Line: Support vector classifier and logistic regression aren't that different!
- Neither they nor any other approach can overcome the "curse of dimensionality".
- SVM uses a non-linear kernel... but could do that with logistic or linear regression too!

# In High Dimensions...

- In SVMs, a tuning parameter controls the amount of flexibility of the classifier.
- This tuning parameter is like a ridge penalty, both mathematically and conceptually. The SVM decision rule involves all of the variables.
- Can get a sparse SVM using a lasso penalty; this yields a decision rule involving only a subset of the features.
- Logistic regression and other classical statistical approaches could be used with non-linear expansions of features. But this makes high-dimensionality issues worse.

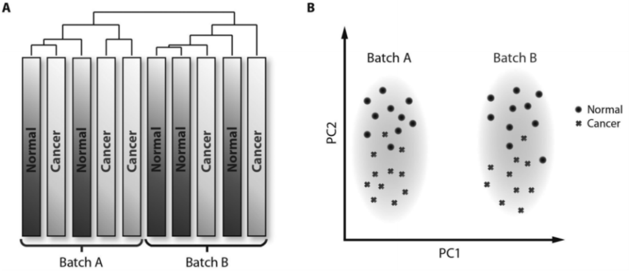
## Let's Try It Out in R!

Chapter 9 R Lab
www.statlearning.com

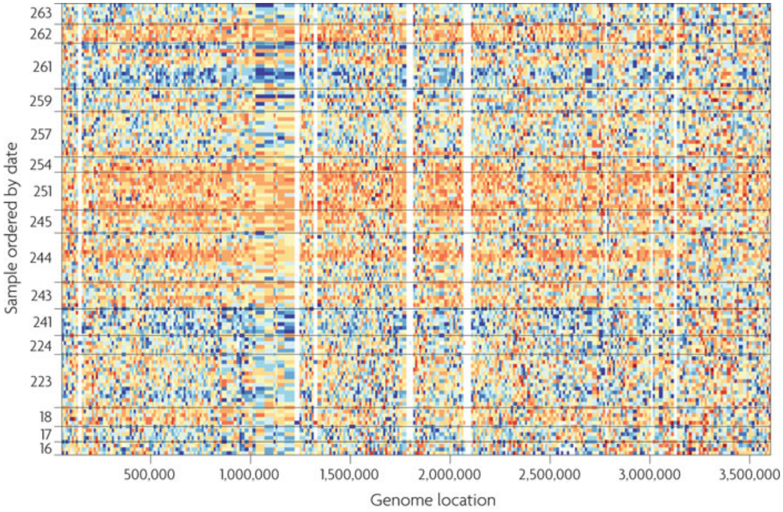
# Batch Effects

- In any sort of omics experiment, need to be very aware of batch effects, induced by non-biological factors such as inter-machine or inter-lab or inter-operator variability, time of day, day of week, position of ceiling fan, ....
- It has been shown many many times that batch effects can be much stronger than biological effects of interest!
- Batch effects can make your data nonsense . . .

# Batch Effects



# Batch Effects in Practice



# Steps to Reduce Batch Effects

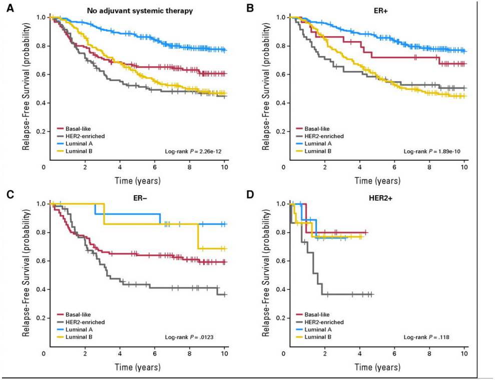
- Randomize sample run times: e.g. don't run cases first and controls second.
- Avoid any extraneous sources of variation, e.g. due to change in person running the experiment.
- It is often better to train a classification or regression method using multiple data sets collected at different institutions, rather than using a single data set.
- Need to validate any results obtained on independent data sets from a different institution.

Batch effects are almost inevitable. But you can do your best to design an experiment and analyze the data in such a way that batch effects do not compromise the results obtained.

# Subtypes of Breast Cancer

- In the past 10 years, global gene expression analyses have identified at least 4 subtypes of breast cancer: Luminal A, Luminal B, Her2-enriched, and basal-like.
- Subgroups differ with respect to risk factors, incidence, baseline prognoses, responses to therapies.
- Want to be able to determine the subtype for a new patient with breast cancer.
- Controversy over the best classifier for this task:
  - PAM50 classifier involves 50 genes.
  - More recent proposal involving three genes.
- Moving target: nobody knows the "true" subtype!
- Prat et al., Breast Cancer Res Treat, 2012

# Why Do We Care About Subtypes?



Citation: Parker et al, Journal of Clinical Oncology, 2009

# Proteomics for Ovarian Cancer

- Ovarian cancer is the leading cause of gynecologic cancer deaths in the USA.
- Much interest in detecting the cancer at an earlier stage.
- In 2002, Petricoin and Liotta – investigators from FDA and NCI – reported in The Lancet that mass spectrometry analysis of circulating serum proteins can be used to discriminate between healthy patients and those with ovarian cancer.
- Great enthusiasm in the popular press and general public.
- Plans were made to begin marketing a test based on the reported diagnostic.

# Not So Fast!!

- Independent researchers took a look at the data, which was publicly available, and discovered:
  - inadvertent changes in protocol mid-experiment: i.e. major batch effects.
  - problems with instrument calibration.
  - difference in processing between tumor and normal samples.
- In summary: the observed differences between cancer and normal proteomic patterns were attributable to "artifacts of sample processing, not the underlying biology of cancer."

# Gene Expression Signatures for Cancer Treatment

- In the early 2000's, Joe Nevins, Anil Potti, and other researchers at Duke University began developing expression-based predictors of response to chemotherapy.
- Many (dozens of!) very promising and very high-profile papers were published in Nature Medicine, The Lancet, Journal of Clinical Oncology, and more.
- Several clinical trials were initiated, using these predictors to direct therapy for cancer patients.
- This research was hailed as a major breakthrough in cancer treatment, and researchers from all over the world tried to use these sorts of techniques in their own labs.

# Upon Closer Inspection....

- Using the fact that some of the data were publicly available, independent researchers discovered the following errors (among many others):
  - Off-by-one errors in gene lists
  - The same heatmap displayed in multiple (unrelated) papers
  - Genes not measured on the array were reported as being part of the predictor obtained, and as providing evidence for biological plausibility
  - Reversal of sensitive/resistant labels
- A shocking paper published by Baggerly and Coombes in Annals of Applied Statistics, detailing all of the errors made: "One theme that emerges is that the most common errors are simple (e.g., row or column offsets); conversely, it is our experience that the most simple errors are common."

# What Went Wrong?

A blasé approach to high-dimensional data analysis:

- Need to have a proper independent test set, that you simply cannot peek at under any circumstances!
- Need to have clearly documented code that contains all steps of the analysis, from start to finish. You must be able to share this code with independent researchers, and you must be confident that your code is correct. If not, then your work isn't ready for prime time.

# The Stakes are High!

At Duke:

- Dozens of papers retracted;
- Careers and reputations ruined;
- Patients endangered through unethical clinical trials.

Plus, a 60 Minutes special feature and an Institute of Medicine Committee!!!