Supervised Learning: Classification, Part I

Noah Simon & Ali Shojaie

July 15-17, 2020
Summer Institute in Statistics for Big Data
University of Washington

Classification

- Regression involves predicting a continuous-valued response.

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

- 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.

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.

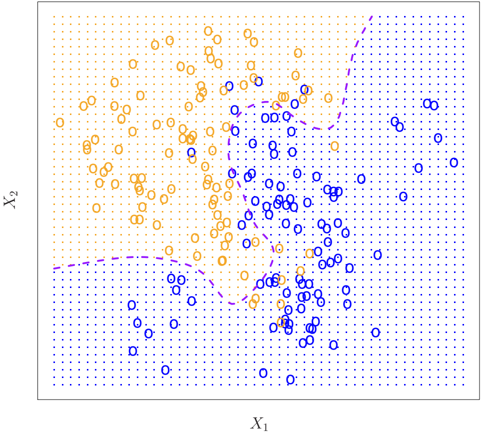
Classification

- There are many approaches out there for performing classification.
- We will discuss 3: k-nearest neighbors, logistic regression, and support vector machines.

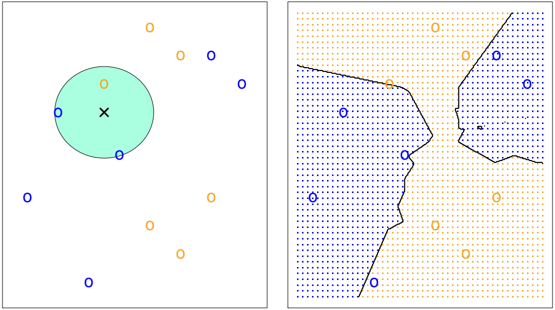
K-Nearest Neighbors

- 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.

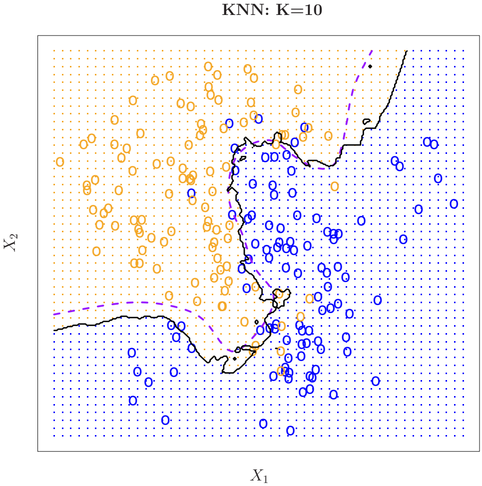
K-Nearest Neighbors



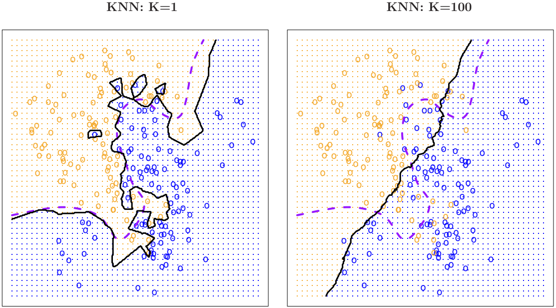
K-Nearest Neighbors



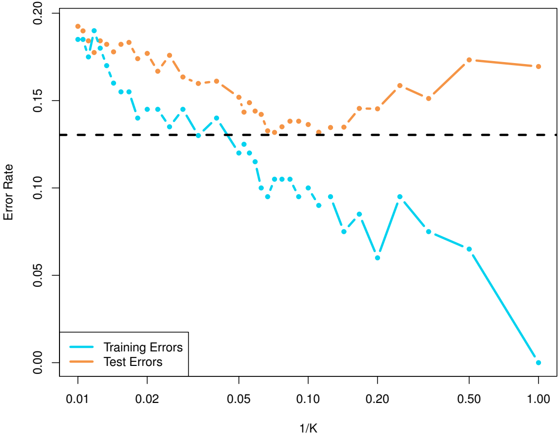
K-Nearest Neighbors



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.

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.

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

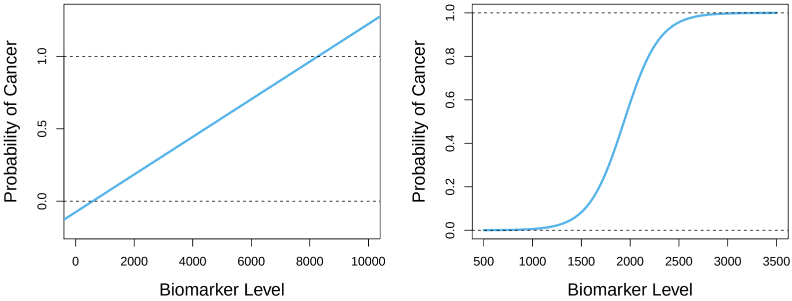
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

- $p(X)$ 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))
```

Three Ways to Extend Logistic to High Dimensions

Three Ways to Extend Logistic to High Dimensions

1. Variable Pre-Selection
2. Ridge Logistic Regression
3. Lasso Logistic Regression

Three Ways to Extend Logistic to High Dimensions

1. Variable Pre-Selection
2. Ridge Logistic Regression
3. Lasso 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:

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

What is an appropriate validation measure?

For probablistic classification

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

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

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

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

- 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

Support Vector Machines

- Developed in around 1995.

Support Vector Machines

- Developed in around 1995.
- Touted as "overcoming the curse of dimensionality."

Support Vector Machines

- Developed in around 1995.
- Touted as "overcoming the curse of dimensionality."
- Does not automatically overcome the curse of dimensionality!!!

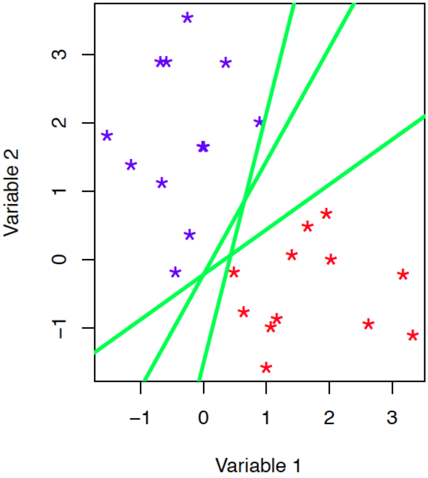
Support Vector Machines

- 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.

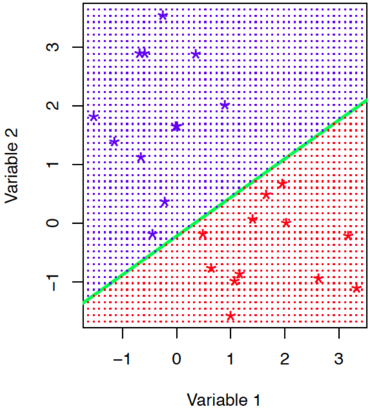
Support Vector Machines

- 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

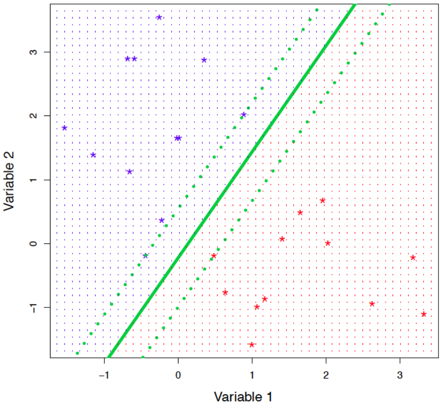


Classification Via a Separating Hyperplane



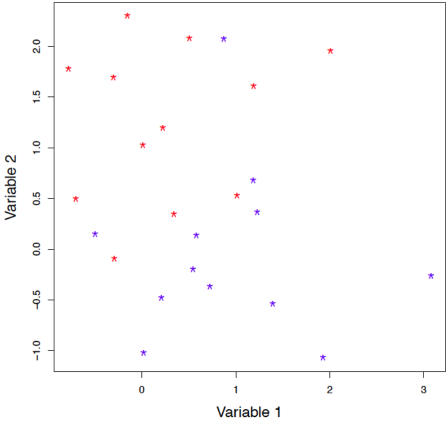
Blue class if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > c$; red class otherwise.

Maximal Separating Hyperplane

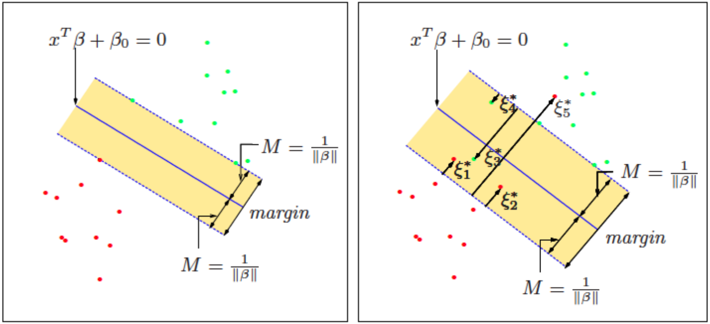


Note that only a few observations are on the margin: these are the support vectors.

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".

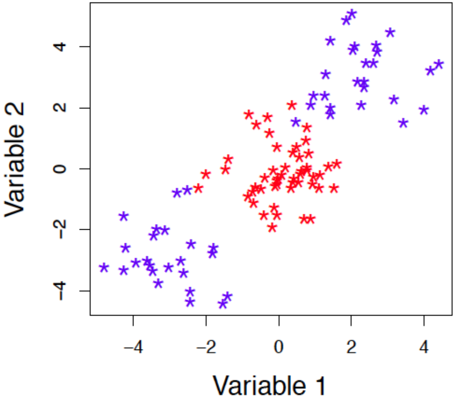
Support Vector Machine

- 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.

Support Vector Machine

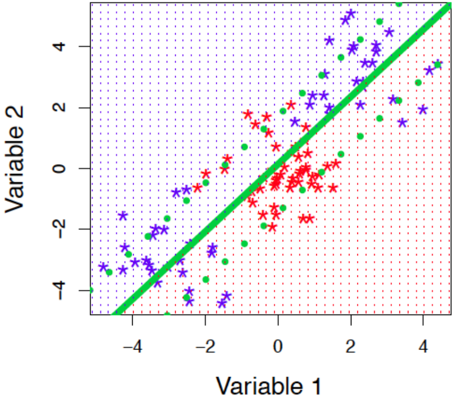
- The support vector machine is just like the support vector classifier, but it elegantly allows for non-linear expansions of the variables: “non-linear kernels”.
- However, linear regression, logistic regression, and other classical statistical approaches can also be applied to non-linear functions 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



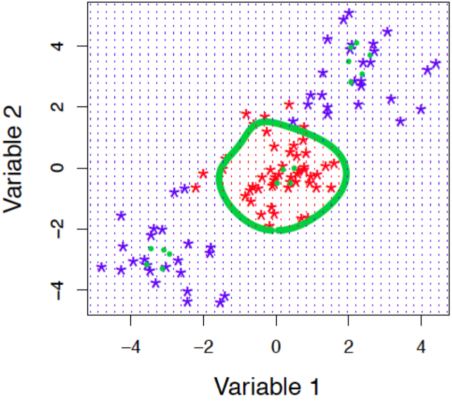
This will be hard for a linear classifier!

Try a Support Vector Classifier



Uh-oh!!

Support Vector Machine



Much Better.

Is A Non-Linear Kernel Better?

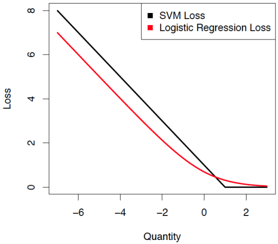
Is A Non-Linear Kernel Better?

- 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.

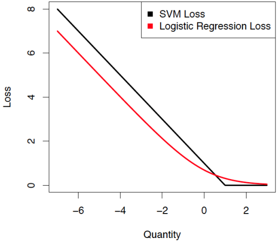
Is A Non-Linear Kernel Better?

- 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

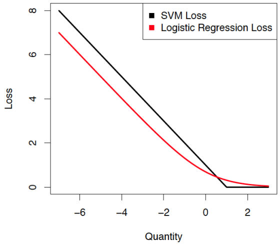


Support Vector Classifier Versus Logistic Regression



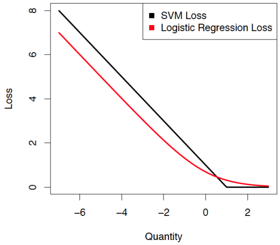
- Bottom Line: Support vector classifier and logistic regression aren't that different!

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".

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 High Dimensions...

- In SVMs, a tuning parameter controls the amount of flexibility of the classifier.

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.

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.

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

Discussion Questions

Suppose someone came to a statistical consulting service you were running and said...

I want to try and classify patients as having breast cancer, or not based on gene expression in serum.

I'm pretty excited because I just found two awesome datasets:

The first, from the Farnsworth Lab, has serum expression measured using RNA-seq in 5000 *patients with breast cancer;*

The second, from the Wernstrom Lab, has serum expression measured using microarrays on 5000 *healthy patients.*

I wanted to combine them to build my classifier

What concerns, if any, come to mind?

Discussion Questions

Suppose we want to classify patients as having cancer/not having cancer using methylation on cf-dna fragments

In particular, say we initially consider 10000 cpg sites, and try to build a classification model that uses proportion of methylated fragments at each of those sites as features.

Would it make sense to run an SVM with a non-linear kernel here?

If we used cross-validation to select between both that SVM, and a LASSO-logistic regression, what might happen?