Tips for Applying Machine Learning

Kamalika Chaudhuri

Topics We Have Not Covered

- Preprocessing Data and Generating Features
- Machine Learning Debugging

Preprocessing Data and Generating Features

Most ML algorithms assume data is a vector of features.

How do we get these features?

Preprocessing Data and Generating Features

Most ML algorithms assume data is a vector of features.

How do we get these features?

Domain Dependent, often needs a lot of expertise

Categorical Data

Data is discrete, and comprises of unrelated categories

Example: States in USA

"Alaska", "Hawaii", "California", etc

Solution:

If there are k categories, convert to 0-1 vector of length k.
Coordinate i of vector is 1 if the category is i, 0 otherwise

Image Data

Simplest: Convert to a vector of pixel colors

More complicated features used for object recognition in computer vision

Text Classification

Most common model: Bag of words

Common Preprocessing Steps:

1. Remove common words, such as pronouns, prepositions, common adjectives, etc

2. Stem or lemmatise words (eg, convert “includes”, “include” and “included” to “include”)

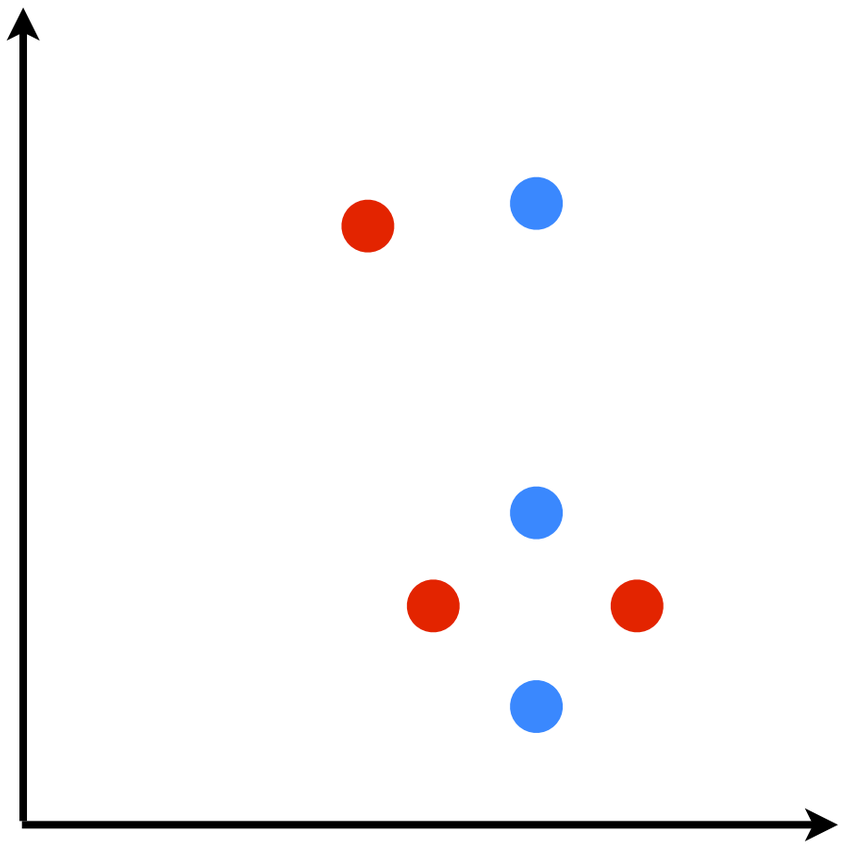
3. Convert to vector of words. Normalize vectors

Other Domains have Domain Dependent Features

Speech, Music: MFCC

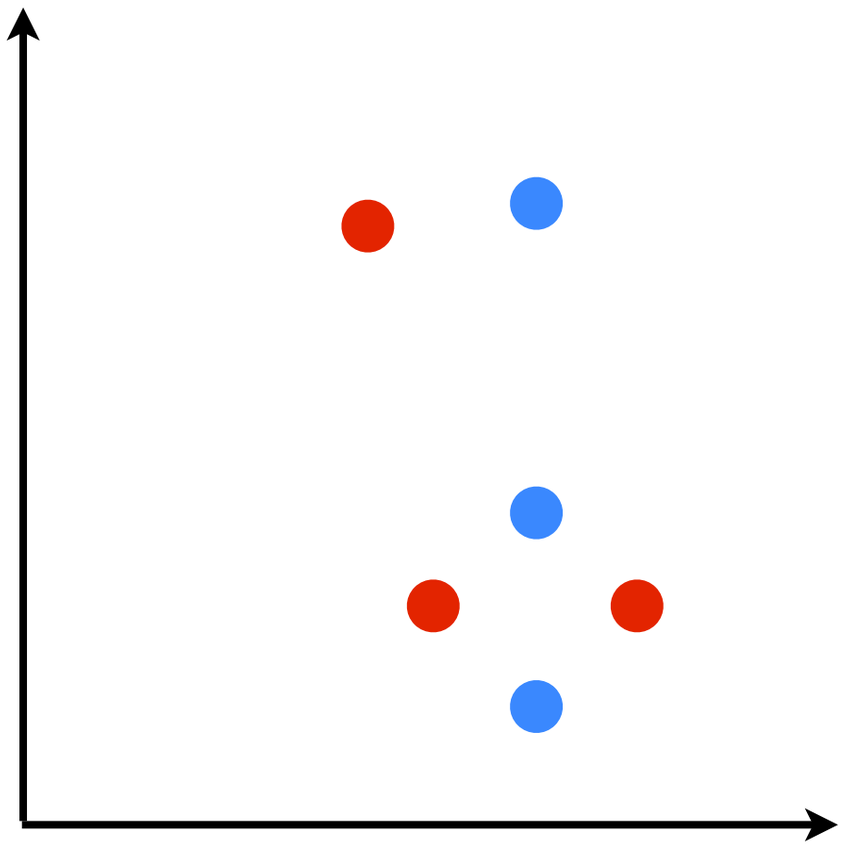
Medical data: etc

Good Features are Important!

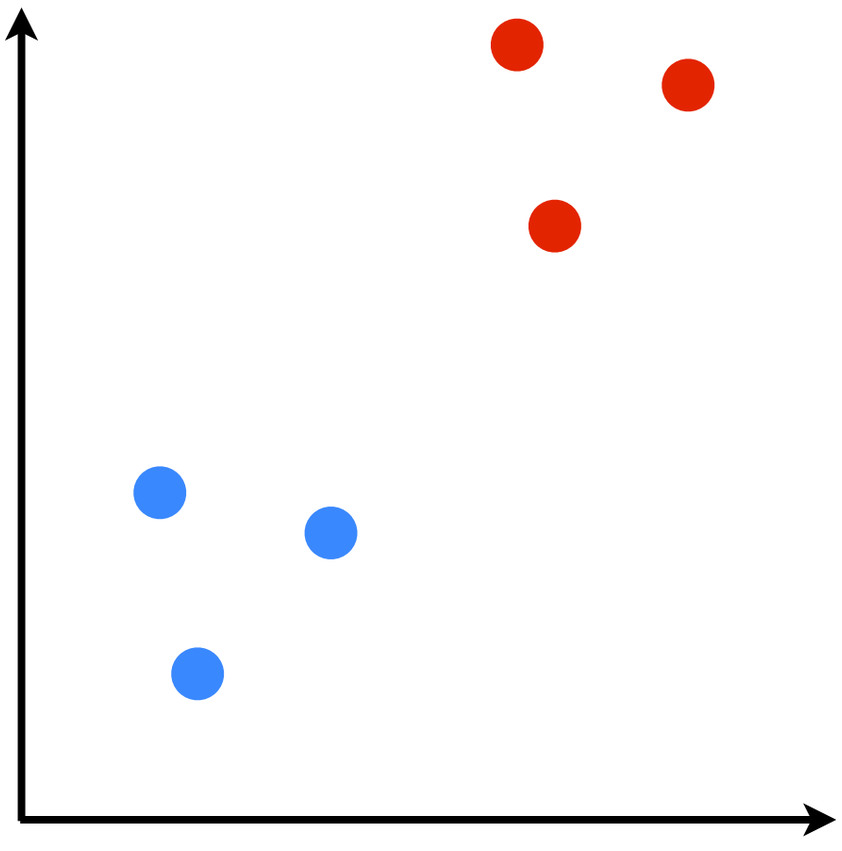


Feature Space 1

Good Features are Important!

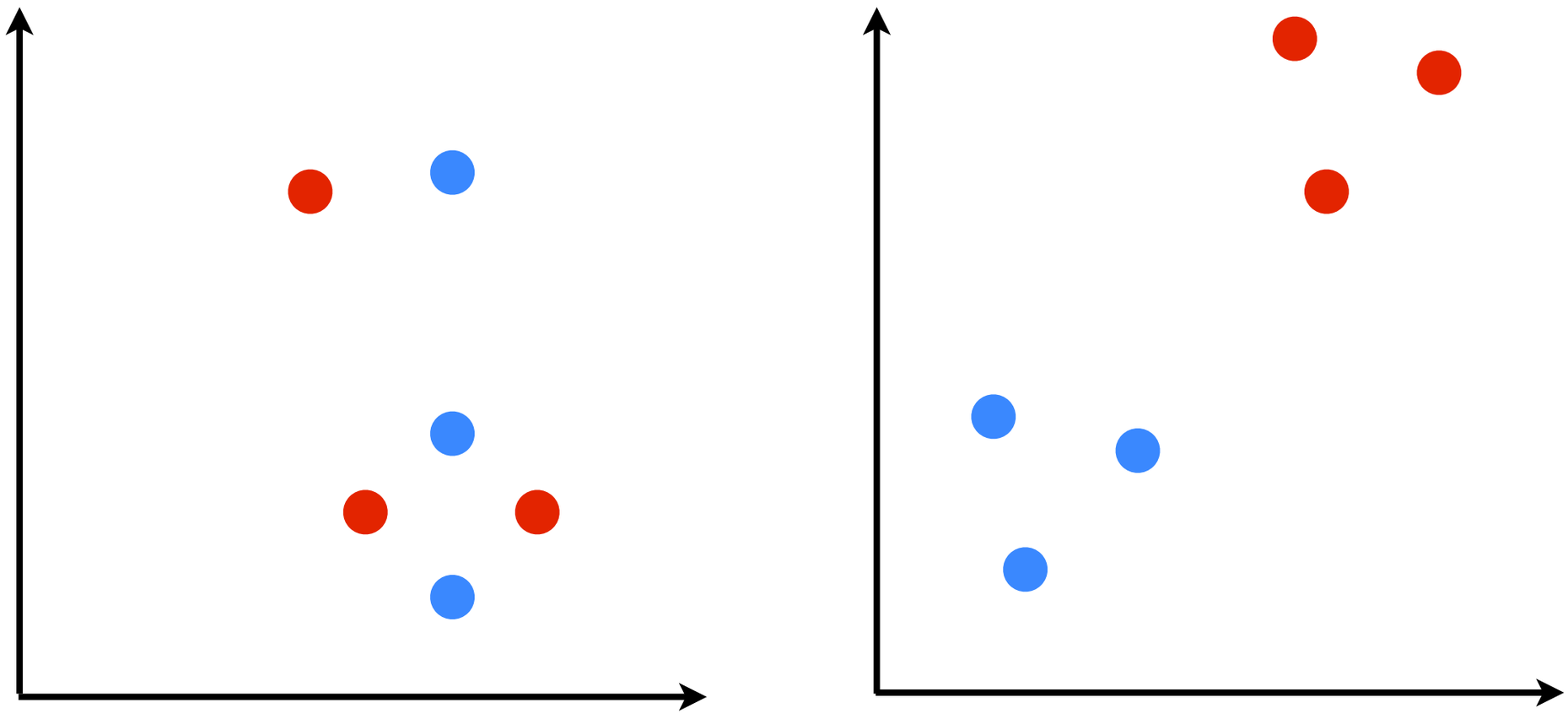


Feature Space 1



Feature Space 2

Good Features are Important!



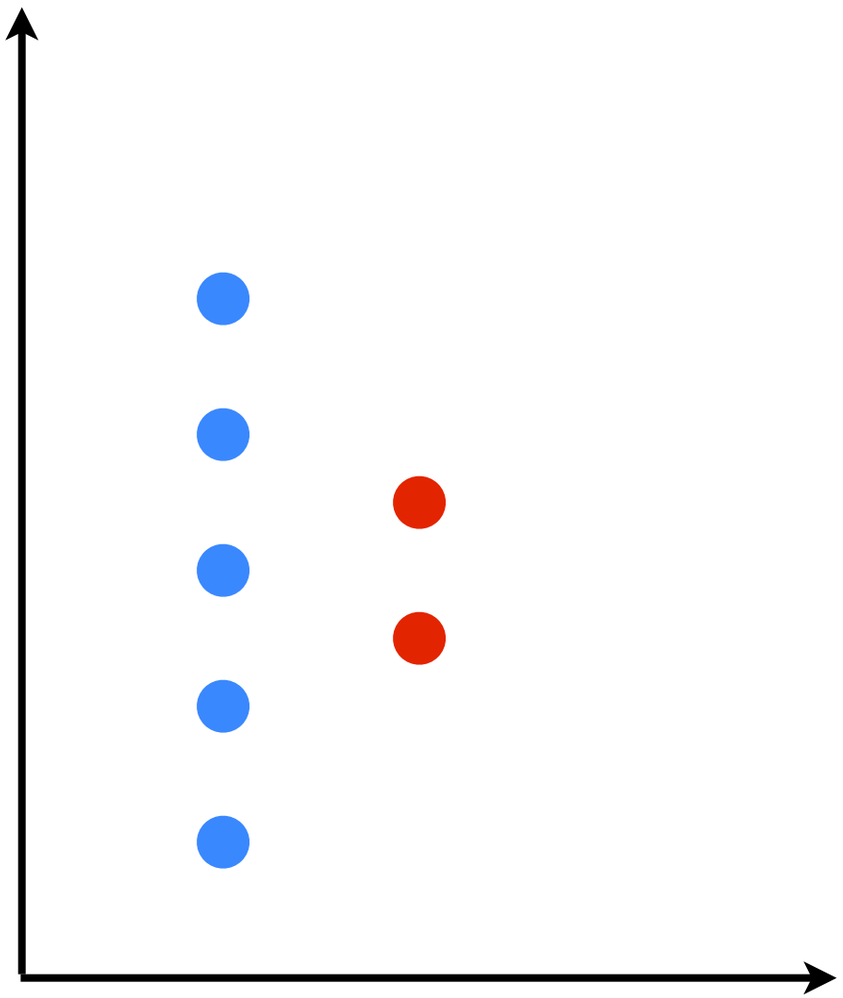
Feature Space 1

Feature Space 2

An algorithm is only as good as the features it has

Combining Multiple Features

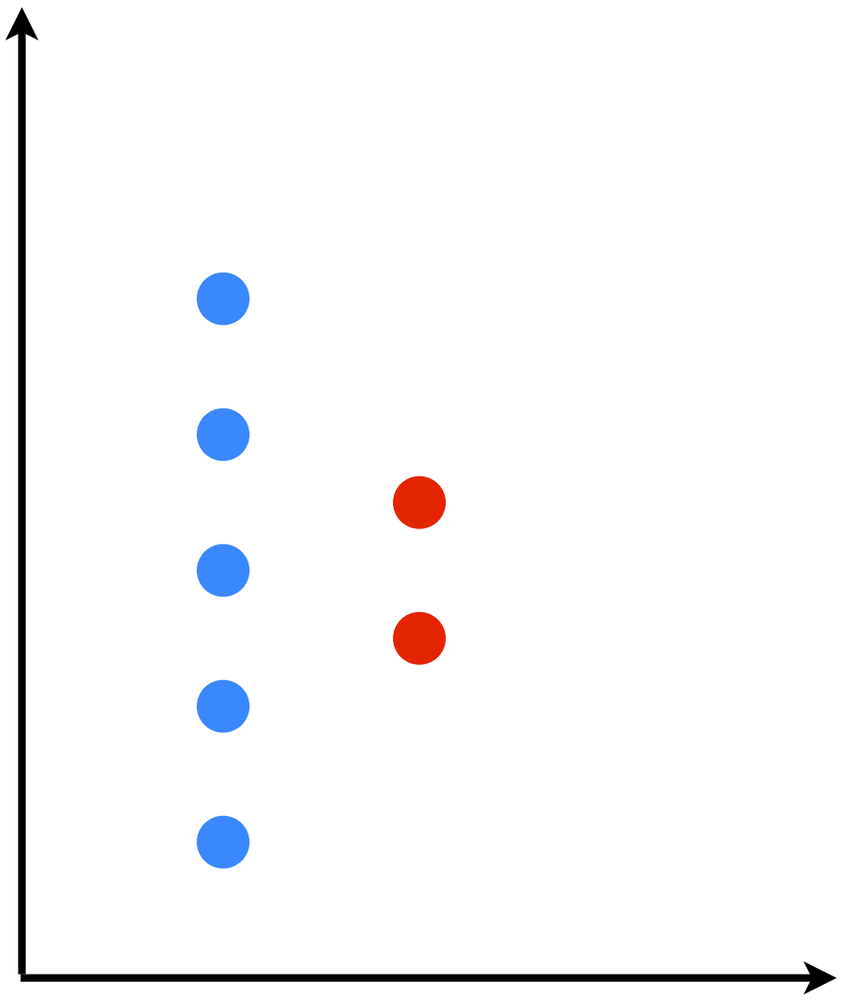
Units matter! (Remember: HW4 Problem 2)



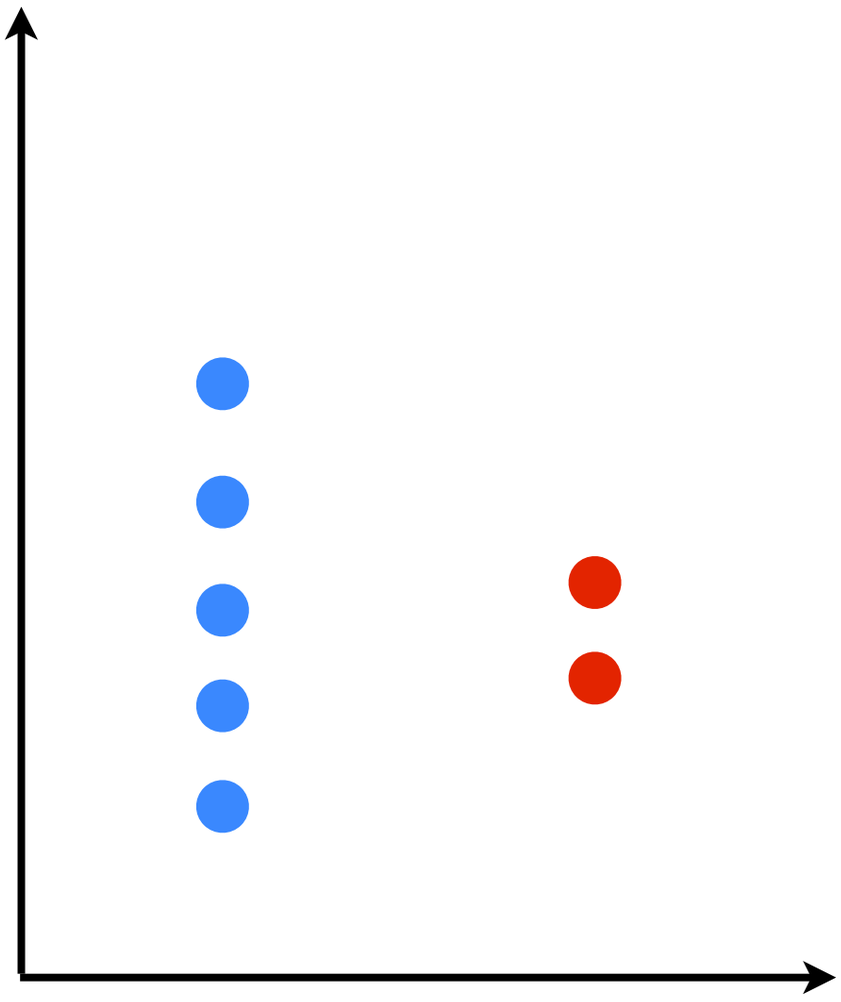
Feature Space 1

Combining Multiple Features

Units matter! (Remember: HW4 Problem 2)



Feature Space 1



Feature Space 2

Combining Multiple Features

Units matter! (Remember: HW4 Problem 2)

If a single feature has a much higher scale than others, then small variations in this feature can dominate the results of the algorithm

Solution: Normalize each feature s.t the max value is 1

ML Debugging

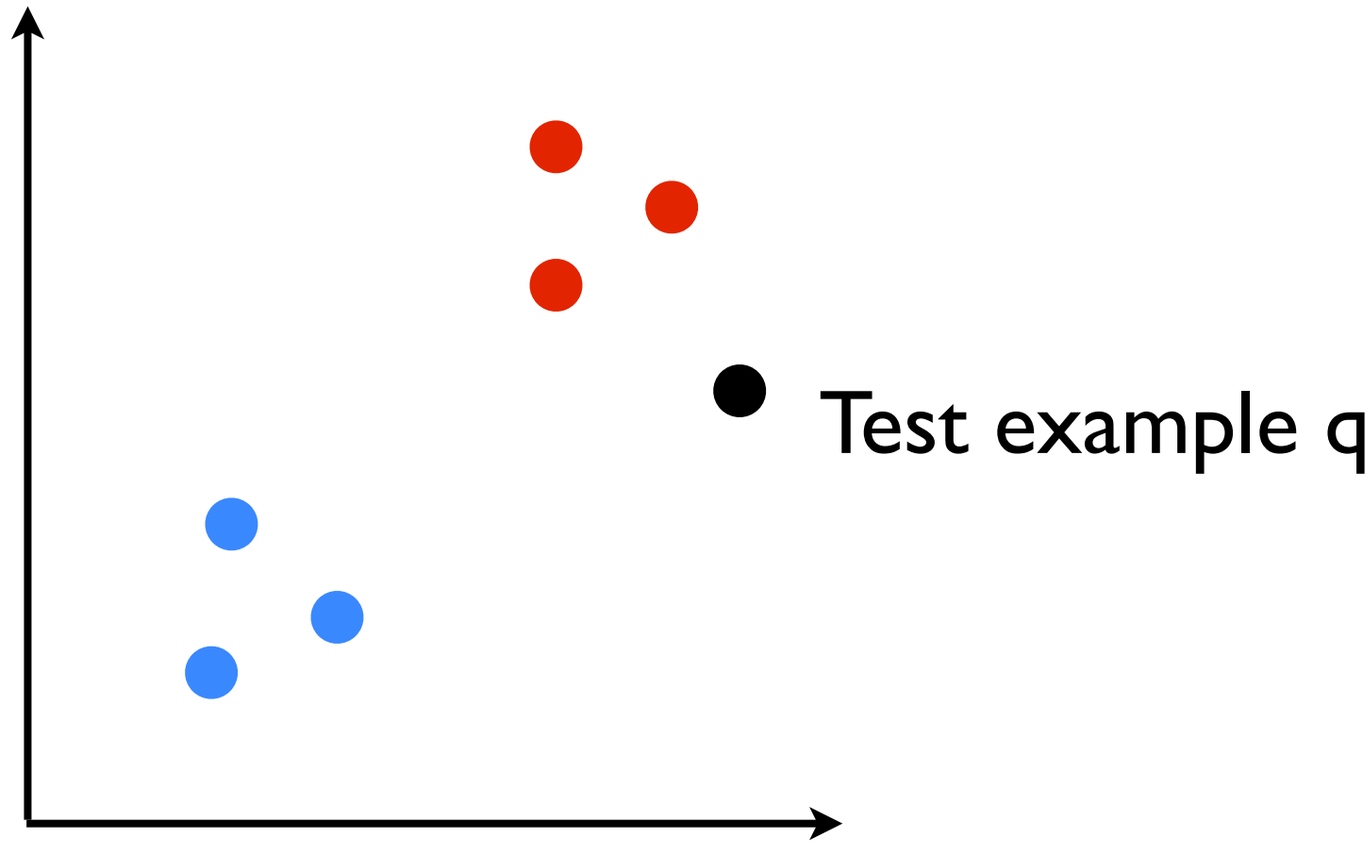
- Debugging your code
- Debugging the machine-learning part

Debugging Code

- Try small examples that you can do by hand
- Visualizations for slightly larger examples

Debugging Code

- Try small examples that you can do by hand
- Visualizations for slightly larger examples



Debugging Code

- Try small examples that you can do by hand
- Visualizations for slightly larger examples
- Visualizing high dimensional data is hard. Try visualizing different projections or visualizing in different feature spaces

Debugging ML

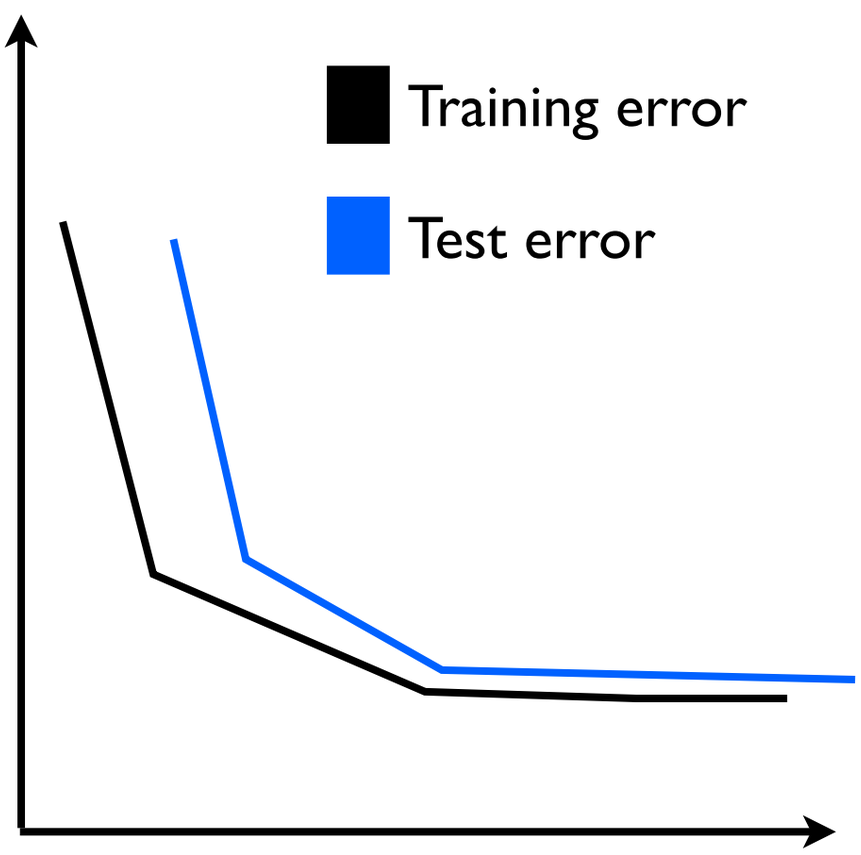
- Bias/Variance Diagnostics
- Effect of the algorithm

Example

Suppose you want to do spam classification. You preprocess the data, pick a set of features, and apply perceptron. You get 25% error.

What could have gone wrong?

Learning Curves



High Bias

High Variance (in the beginning)

How to decrease bias?

1. Choose more features (more words from email)
2. Choose email header features
3. More general problems: pick a kernel (polynomial or Gaussian) -- not as suitable for text

How to decrease variance?

1. Choose less features (check to see which words are more "correlated" with the labels or occur more frequently and keep them)

2. Get more samples

Another Issue: The Algorithm

Maybe you used a single pass of perceptron and perceptron has not converged, so we are working with a suboptimal classifier

Solutions:

1. Try running perceptron for more passes and see if things improve
2. Use a different algorithm -- voted or averaged perceptron, or Support Vector Machines

ML Diagnostics

- Usually problem-specific. You have to use your ingenuity and experience to come up with a diagnostic
- Debugging can be very subtle
- **Tip:** Try to use a good visualization. Again need to use ingenuity, but visualizations help