CS 383 – Machine Learning

Classification – Intro

Slides adapted from material created by E. Alpaydin
Prof. Mordohai, Prof. Greenstadt, Pattern Classification (2nd Ed.),
Pattern Recognition and Machine Learning

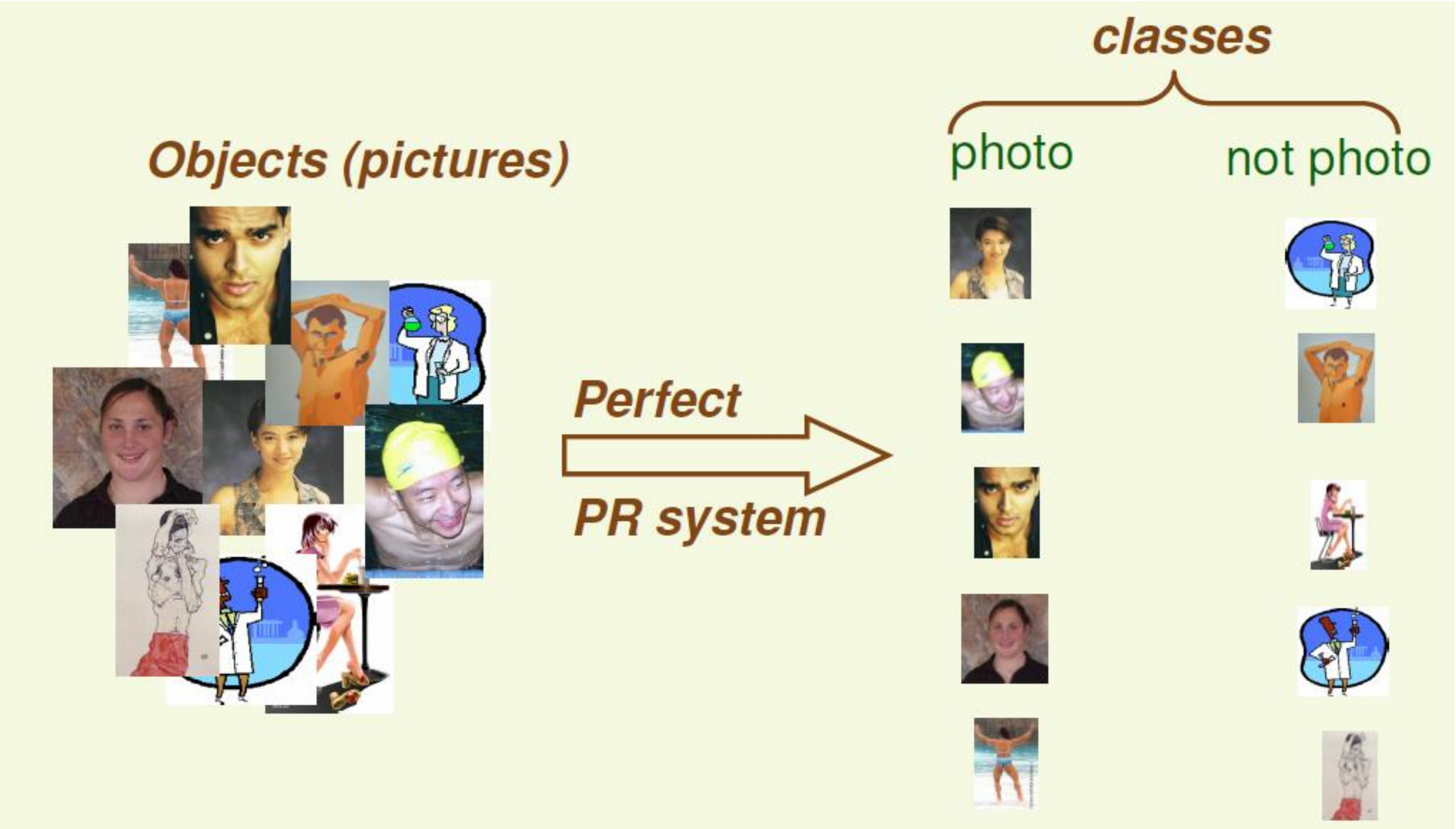
Objectives

- Classification Intro
- Evaluating a Classifier

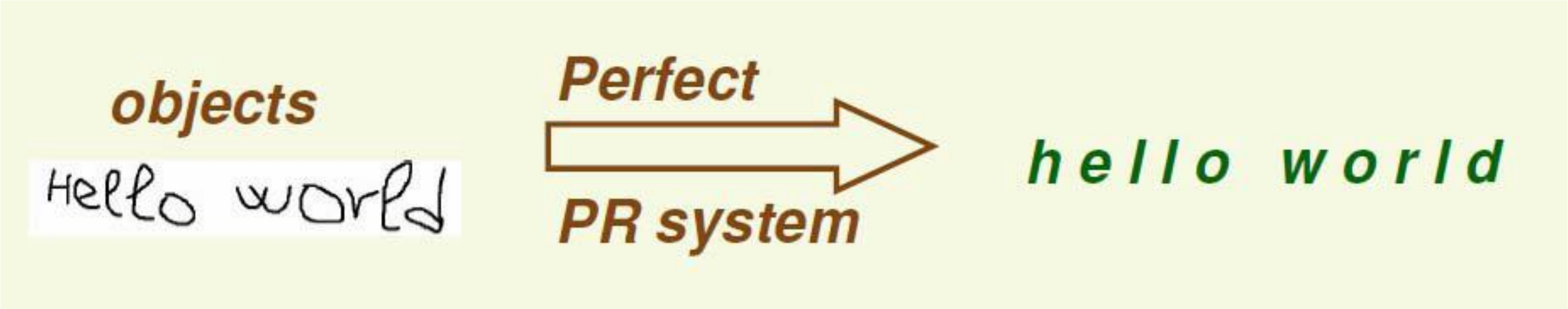
Classification

- Ok so in the course we want to:
  - ~~Look at and manipulate data~~
  - ~~Be able to find patterns in unlabeled data (clustering)~~
  - ~~Build a system that, given some data, can predict a values (regression)~~
    - ~~This system will be built using prior labeled data~~
  - Build a system that, given some data, can predict a label for that data (classification)
    - This system will be built using prior labeled data

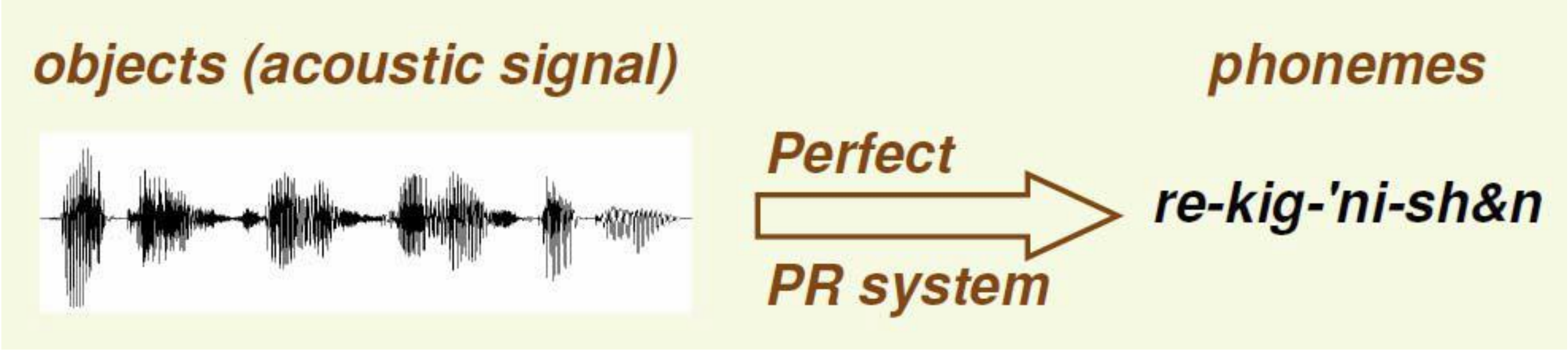
Example: Photograph or Not?



Example: Character Recognition



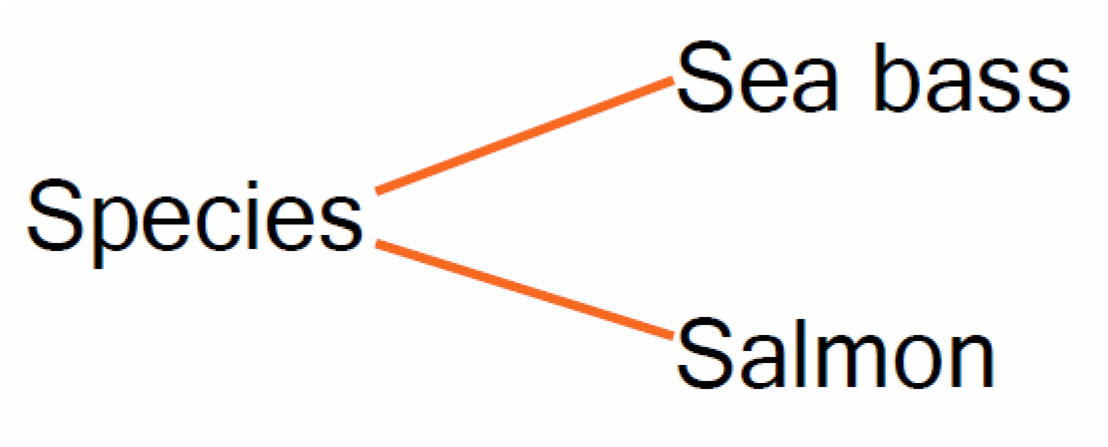
Example: Speech Understanding



Basic Example

An Example

- Classify fish according to species using optical sensing



Training

- Set up a camera and take some sample images
  - Label these images by hand

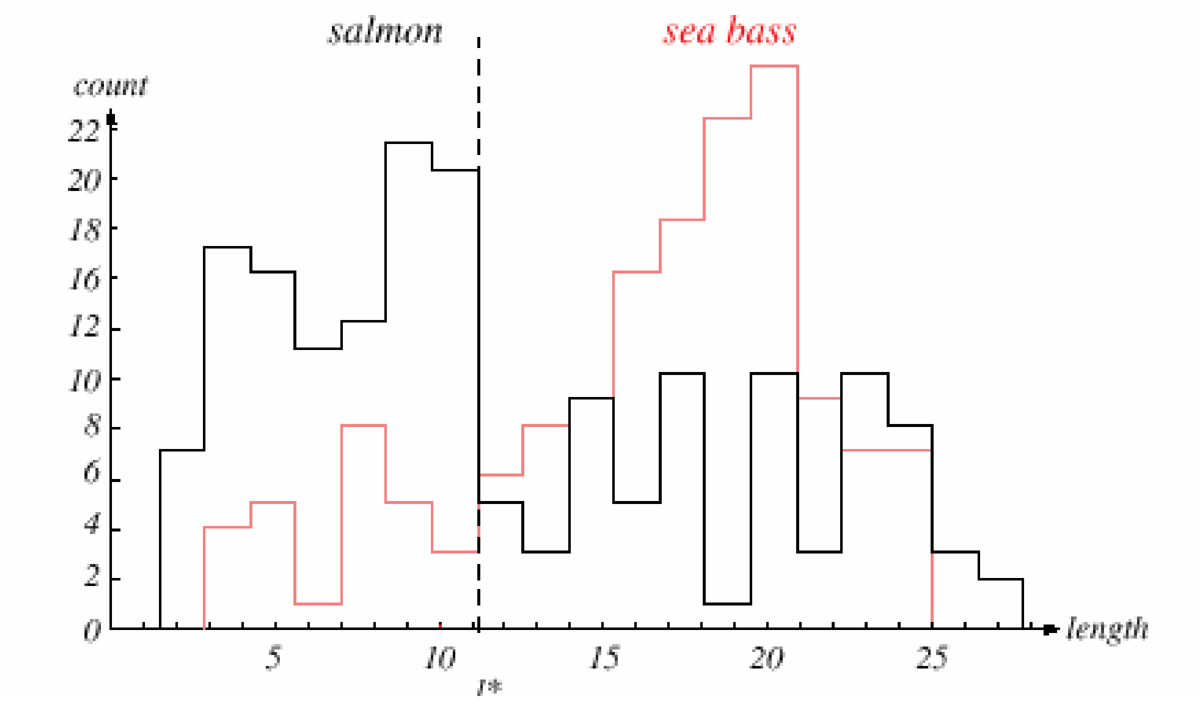
- Extract features
  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth
  - Etc..
- Test whether this set of features is useful for a classifier

Preprocessing

- Use a segmentation operation to isolate fish from one another and from the background
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain quantities
- Features are passed to a classifier

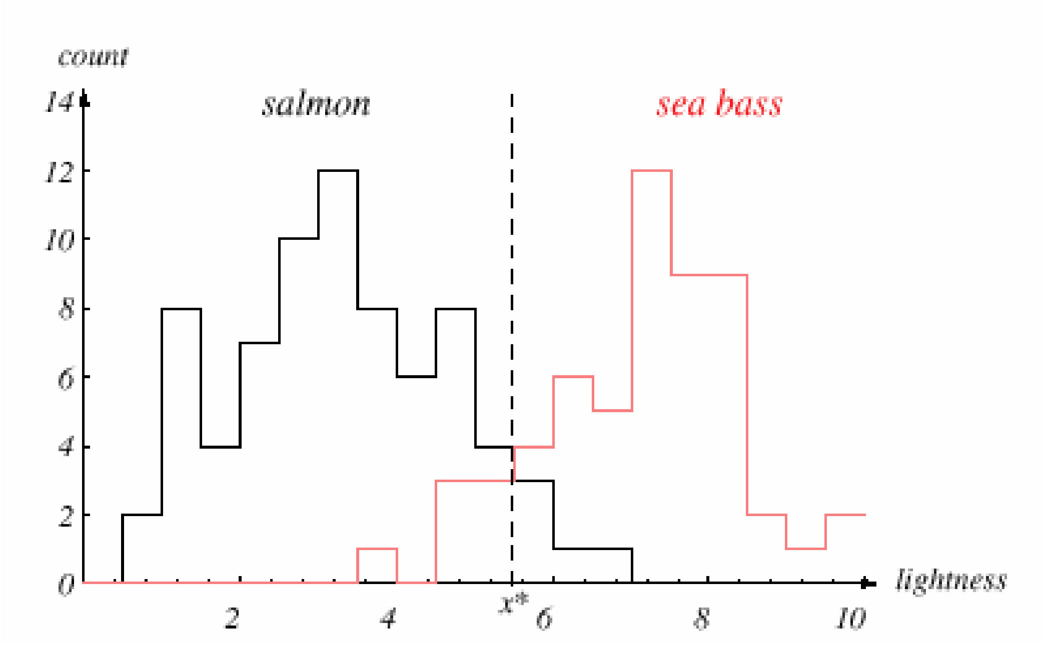
Classification

- Select the length of the fish as a possible feature for discrimination



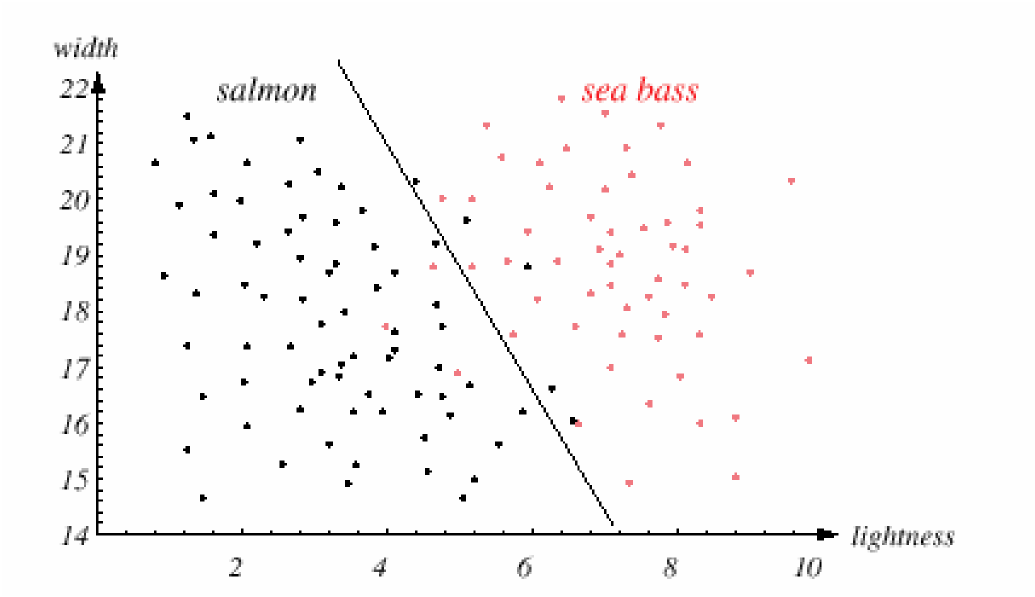
Preliminary Results

- Length alone looks to be a poor feature!
  - About 20% misclassification rate at best threshold choice
- Select lightness as a possible feature



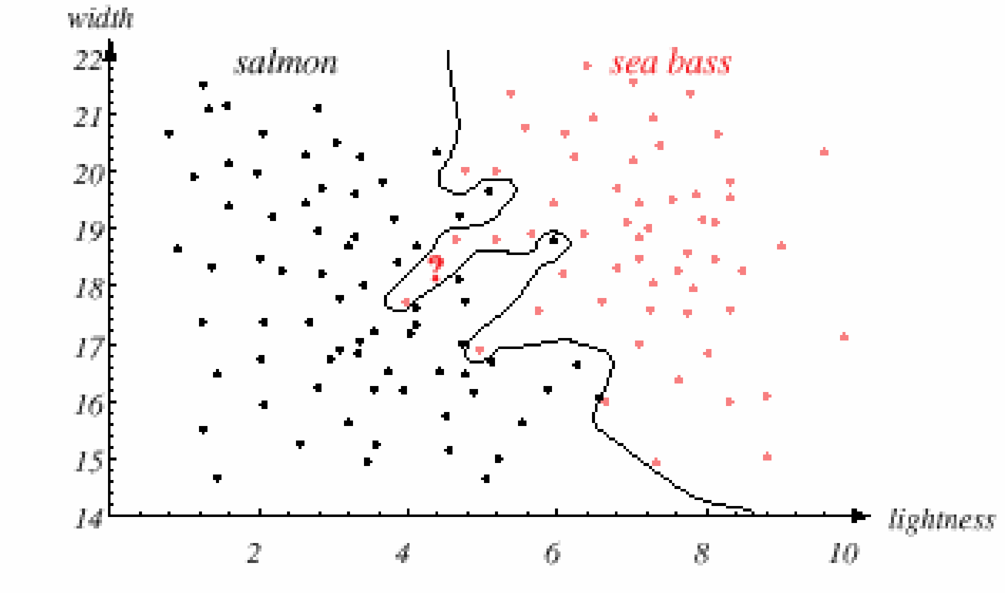
New Classifier

- Use the lightness and add the width of the fish
- Fish
  - $X_{:i} = [X_{:1}, X_{:2}]$
  - Where $X_{:1}$ is the lightness and $X_{:2}$ is the width



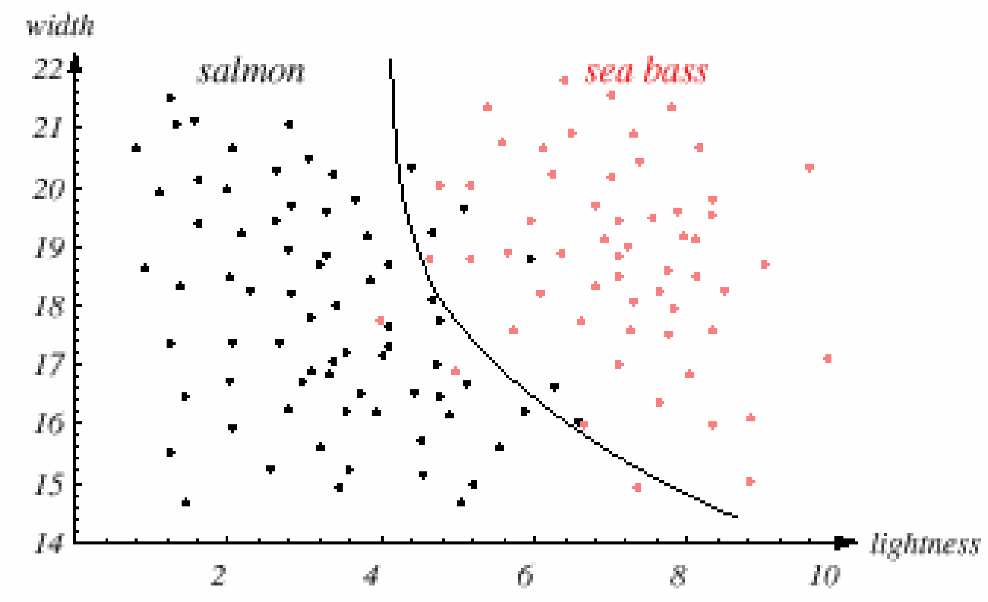
Keep Going…?

- We can add other features that are not correlated with the ones we already have
- Intuitively, the best decision boundary should be the one which provides an optimal performance

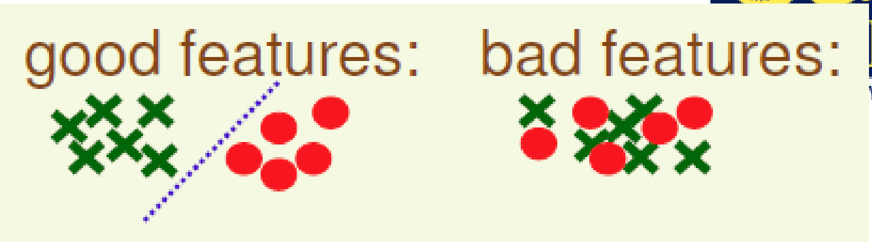


Generalization

- However, the primary goal of designing a classifier is to correctly classify novel input (new instances)
- This is the issue of generalization
- Maybe this generalizes better

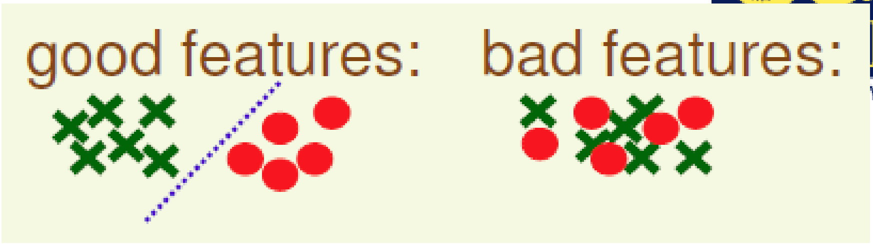


The Design Cycle



- Data collection
  - How do we know when we have enough data for testing and training?
- Feature extraction/choice
  - What features are good?
    - Depends on the problem domain
    - Ideally simple to extract, invariant to irrelevant transformations, insensitive to noise
- Model choice
  - What type of classifier to use?

The Design Cycle



- Training
  - Change/learn parameters of classifier so that the model fits the collected data well
- Evaluating
  - Measure system performance
  - Avoid overfitting
- Computational complexity
  - What is the trade-off between computational ease and performance?
  - How does the choices made scale?

Evaluating a Classification Algorithm

Evaluation

- How can we tell if our ML algorithm is doing well?
- For estimating values (like in linear regression) we can just compute statistics on the error
- How about with classifiers?
  - We can count how often we predict the correct class
- What if the classifier returns a likelihood instead of a discrete class ID?
  - Choose a threshold
  - Anything below that threshold is class 0, anything above it is class 1

Spam Example

- Let’s imagine creating a classifier that detects spam
- It’s easy to catch 100% of spam
  - Throw out ALL the mail
  - Set “threshold” to 0 (or 1)
- It’s easy to make no mistakes on good mail
  - Keep ALL the mail
- Perhaps a good starting point is to choose 50% threshold
  - Anything above this is a “positive hit”
  - Anything below this is a “negative rejection”

Error Types

- True positive = Hit
- True negative = Correct rejection
- False positive = False Alarm (Type 1 error)
- False negative = Miss (Type 2 error)

	Predicted positive	Predicted negative
Positive examples	**True positives**	**False negatives**
Negative examples	**False positives**	**True negatives**

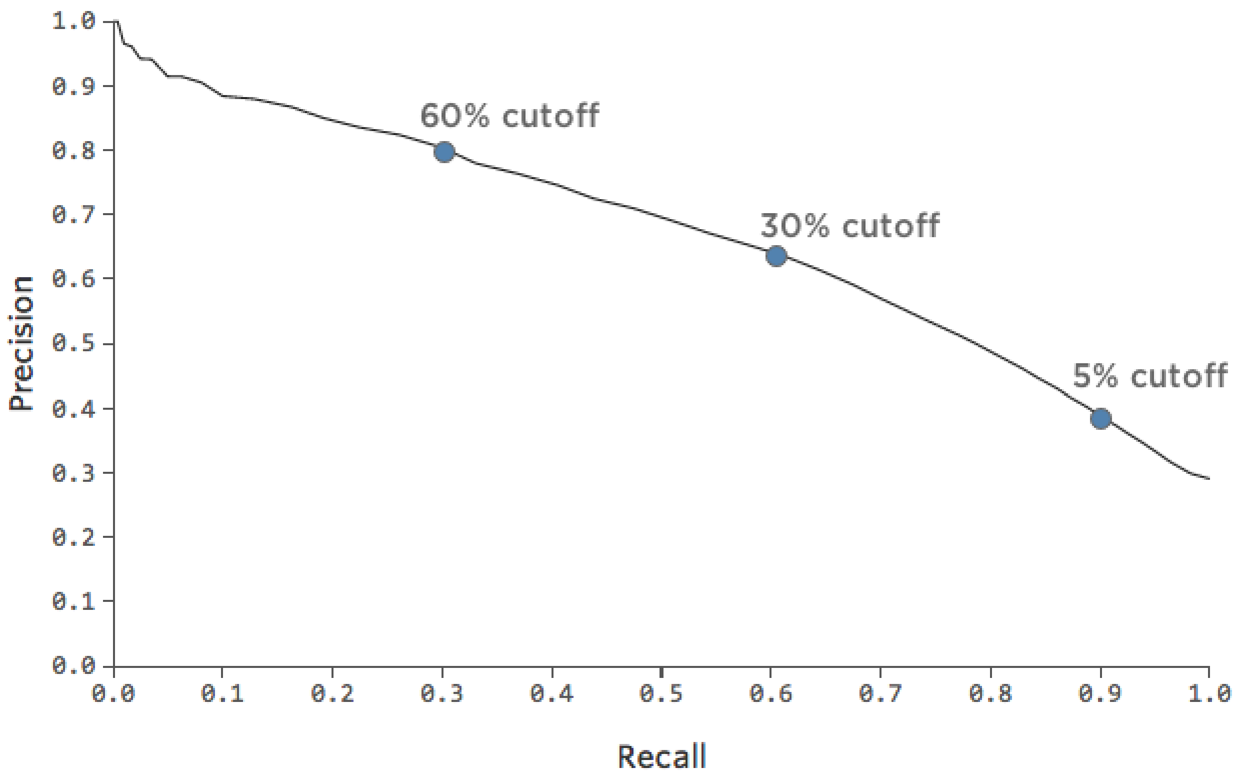
Evaluating your Classifier

- There are several different things we can measure to evaluate the quality of a classifier
  - Depending on the application one may be more important than another
- *Precision* – percentage of things that were classified as positive and actually were positive
  - $Precision = \frac{TP}{TP+FP}$
- *Recall* – the percentage of true positives (*sensitivity*) correctly identified
  - $Recall = \frac{TP}{TP+FN}$
- *f-measure* – The weighted harmonic mean of precision and recall
  - $F_1 = \frac{2*precision*recall}{precision+recall}$

Precision/Recall Tradeoff

- We can vary it by adjusting thresholds
- Plotting precision and recall as a function of the threshold creates something called a *precision-recall* curve (PR)

Precision/Recall Curve



Evaluating your Classifier

- Precision, Recall, and the f-measure are primarily concerned the detection of a particular class (the "positive" label)
- Often we're more interested in getting both classes of a binary classifier correct
  - True positive and true negative

Evaluating your Classifier

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Related to this, we sometimes will look at the true positive rate vs the false positive rate
  - The true positive rate is

$$TPR = Recall = \frac{TP}{TP + FN}$$

  - The false positive rate is

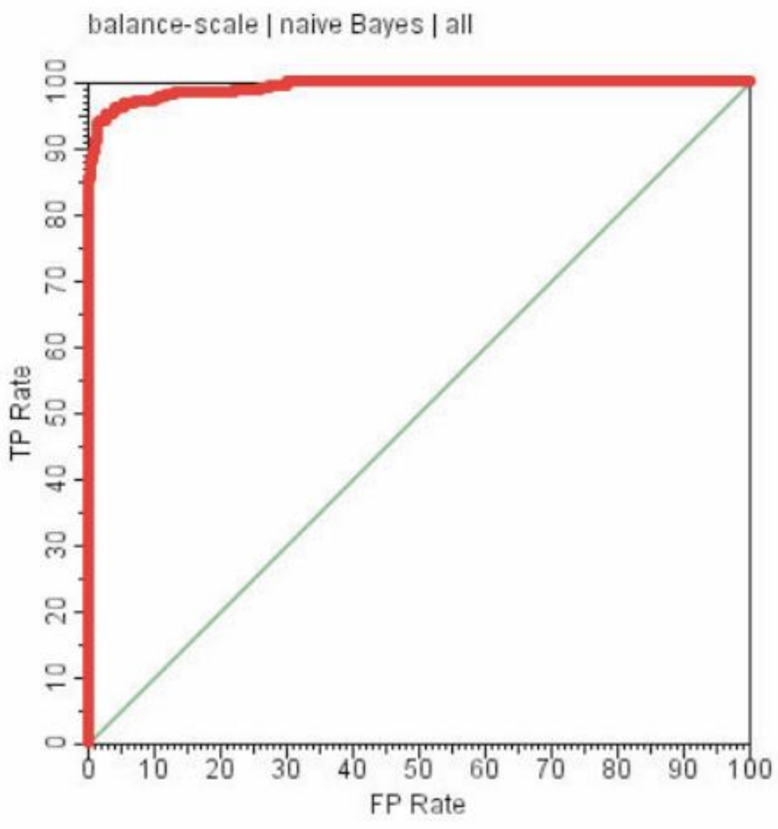
$$FPR = \frac{FP}{FP + TN}$$

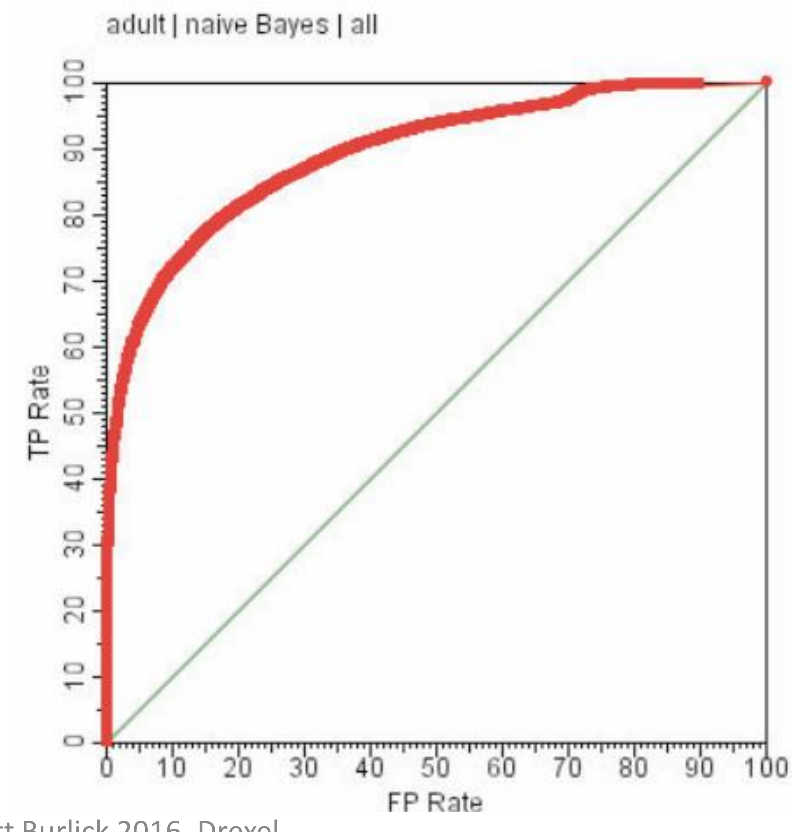
ROC Curves

- Likewise we can plot TPR vs FPR
  - This is often called a receiver operating characteristic (ROC) curve
- Like with PR-Curves, this can be created by varying the parameters that decides positive vs negative classification
- Dominant curves (above and to the left)

Example ROC Curves

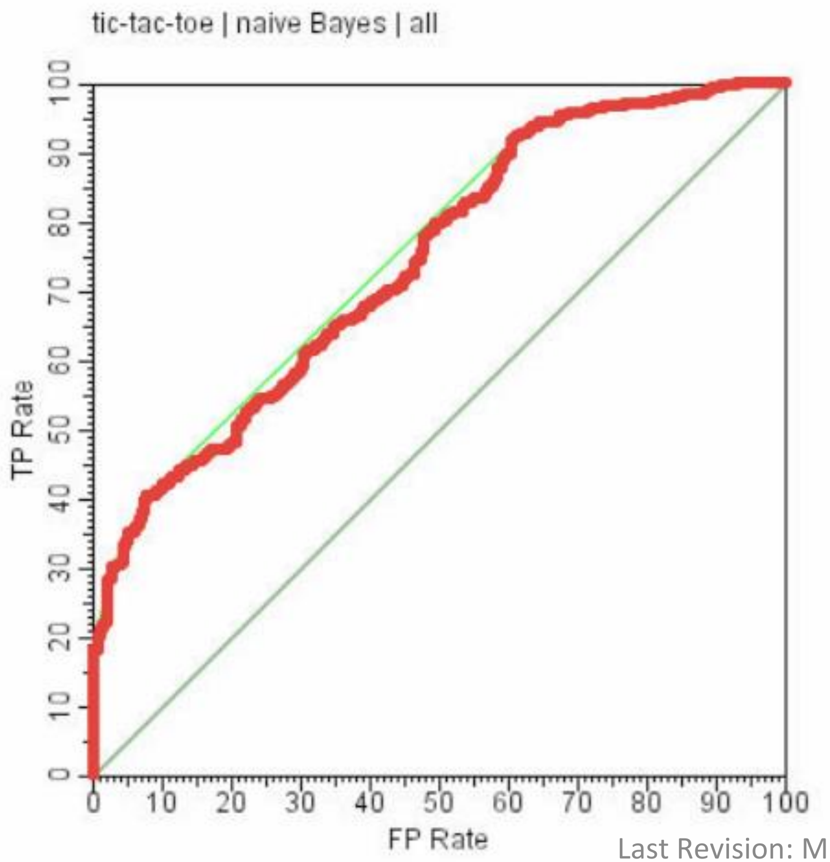
- Good separation, convex

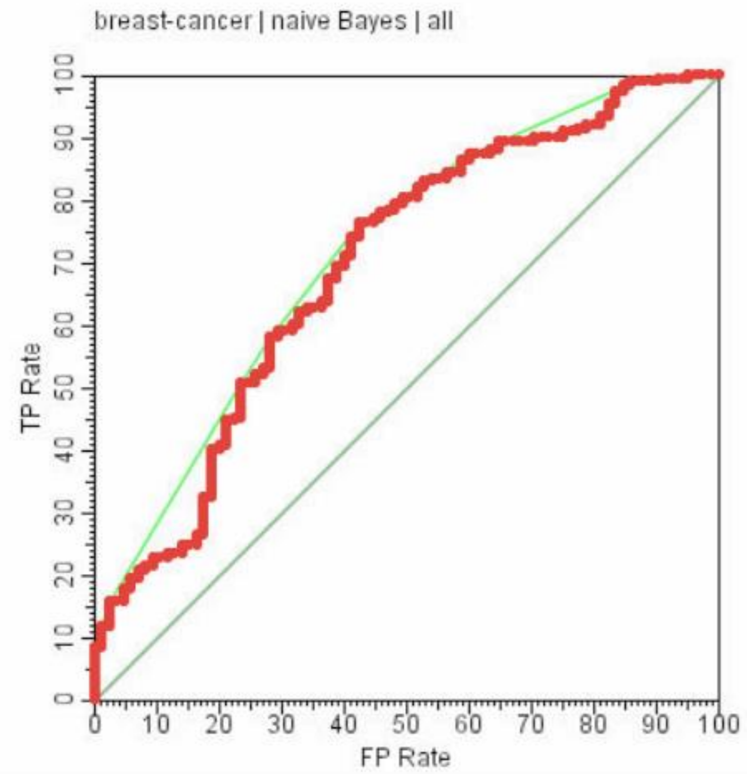




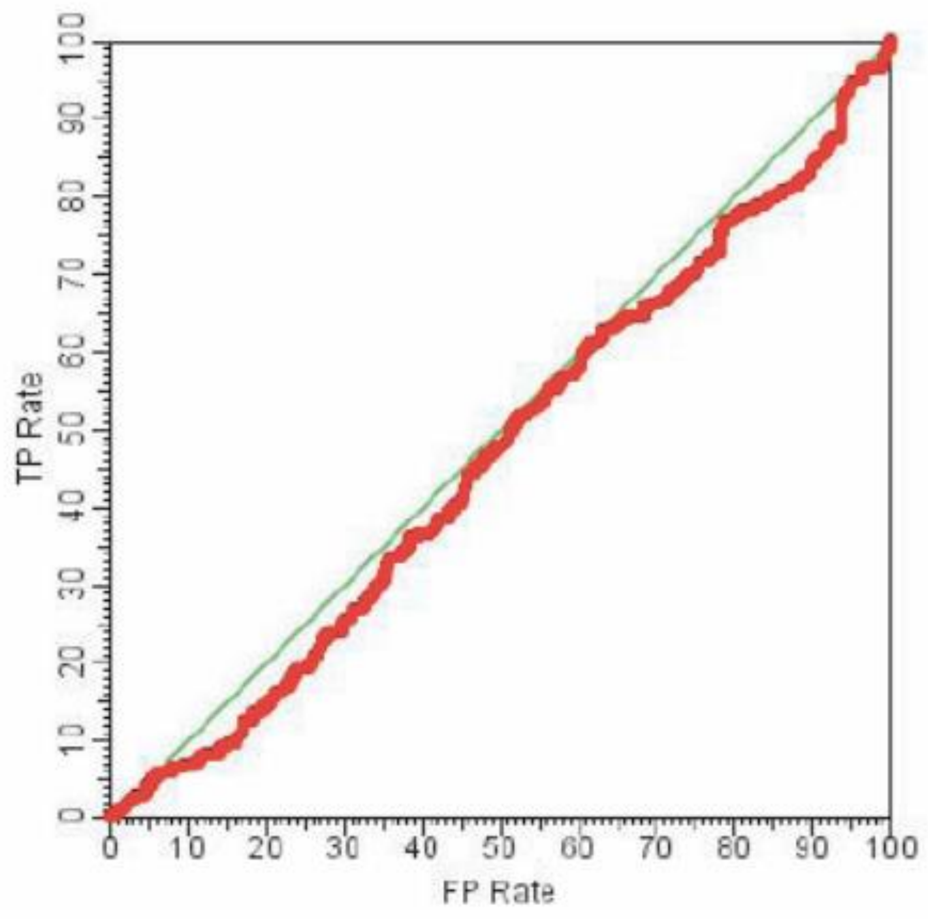
Example ROC Curves

- Poor separation

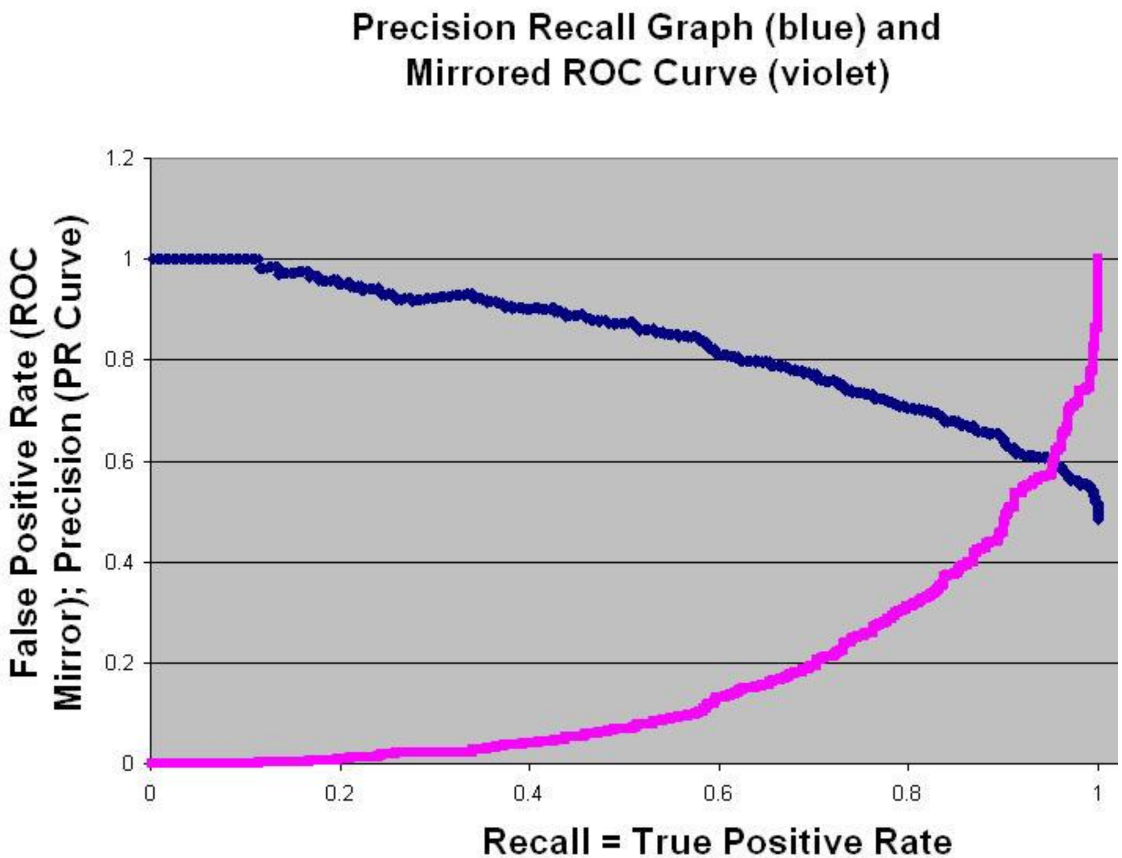




Random



PR and ROC Graphs



Precision/Recall or ROC?

- So which should you use?
  - Depends
- Precision- Recall
  - $Precision\ = \frac{TP}{TP+FP} = P(Y=1|\hat{Y}=1)$
  - $Recall\ = \frac{TP}{TP+FN} = P(\hat{Y}=1|Y=1)$
  - Conditioned on both the true label, Y and the estimated label $\hat{Y}$.
  - As a result the probability of $Y=1$, $P(Y=1)$ will effect results.
    - So use if your specifically care about “this” population/label distribution.
  - Typically use if the “positive” class is more interesting than the “negative” class.
- ROC
  - $TPR = Recall = \frac{TP}{TP+FN} = P(\hat{Y}=1|Y=1)$
  - $FPR = \frac{FP}{FP+TN} = P(\hat{Y}=0|Y=0)$
  - Conditioned only on the actual/true labels
    - So independent of what $P(Y=1)$ actually is
  - So use if you want a more general view independent of label distribution.