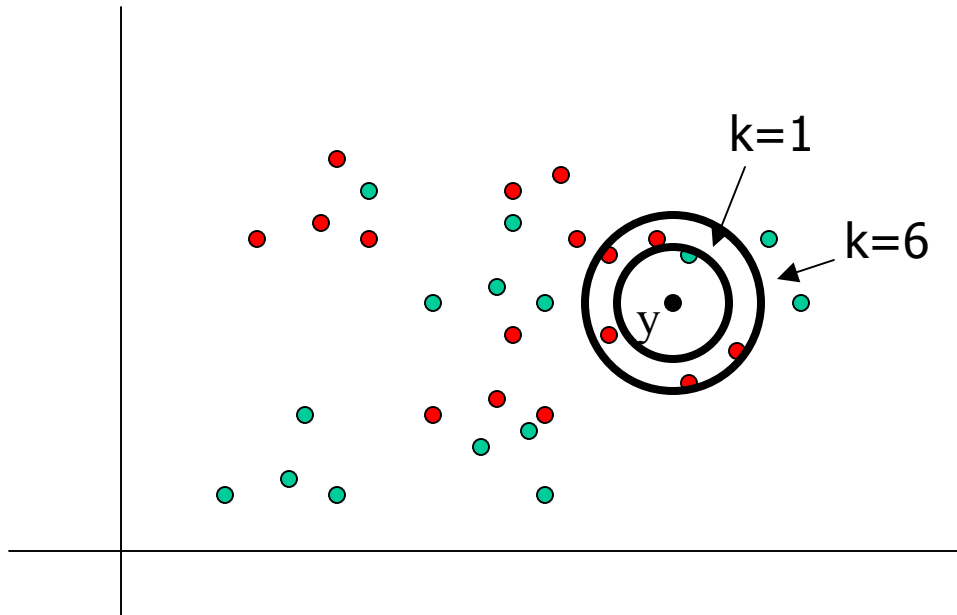


- Today's Reading:
 - HMS, 6.3, 7.3.1, 10
- Today's Lecture:
 - Predictive Modeling
 - bias and variance in KNN
 - SVMs
 - evaluating predictive models
 - feature selection
 - Finding Patterns
- Upcoming Due Dates:
 - H3 due thu

Nearest Neighbor Methods

- To classify a new input vector y , examine the k -closest training data points to y and assign the object to the most frequently occurring class



Issues

- Distance measure
 - Most common: euclidean
- Choosing k
 - Increasing k reduces variance, increases bias
- For high-dimensional space, problem that the nearest neighbor may not be very close at all!
- Memory-based technique. Must make a pass through the data for each classification. This can be prohibitive for large data sets.
- Indexing the data can help; for example KD trees

Bias and Variance in NN vs. Linear Regression

- NN – no assumptions about the model (low bias)
prediction not stable (high variance)

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- LR – strong assumptions about the model (high bias)
prediction is stable (low variance)

$$\hat{y} = X^T \hat{\beta}$$

SVM Resources

- www.support-vector.net
- tutorials:
 - <http://www.support-vector.net/icml-tutorial.pdf>
 - C. J. C. Burges. [A Tutorial on Support Vector Machines for Pattern Recognition](#). *Knowledge Discovery and Data Mining*, 2(2), 1998.
 - A. J. Smola and B. Schölkopf. [A Tutorial on Support Vector Regression](#). NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
- books:
 - Cristianini, N., Shawe-Taylor, J., [An Introduction to Support Vector Machines](#), Cambridge University Press, (2000).
 - Schölkopf, S., Burges, C. J. C., Smola, A. J., [Advances in Kernel Methods: Support Vector Learning](#), MIT Press, Cambridge, MA, (1999).

Support Vector Machines (SVMs)

- currently the 'en vogue' approach to classification, regression
- successful applications in a wide variety of fields: bioinformatics, text, handwriting recognition, image processing
- draws people from machine learning, optimization, statistics, neural networks, functional analysis
- introduced by Bosner, Guyon and Vapnik, 1992
- Kernel Machines: SVMs are a particular instance

Basic Idea

- For a given learning task, with a given finite amount of training data, the best generalization performance: balance between the accuracy on that particular training set and the “capacity” of the machine (ability of machine to learn any training set without error).
 - botanist example (Burges)
 - **Too much capacity** is like a botanist with a photographic memory who, when presented with new tree, concludes that it is not a tree because it has a different number of leaves from anything seen before
 - **Too little capacity**: if it’s green, it’s a tree
- ... neither generalize well!

Linear SVMs

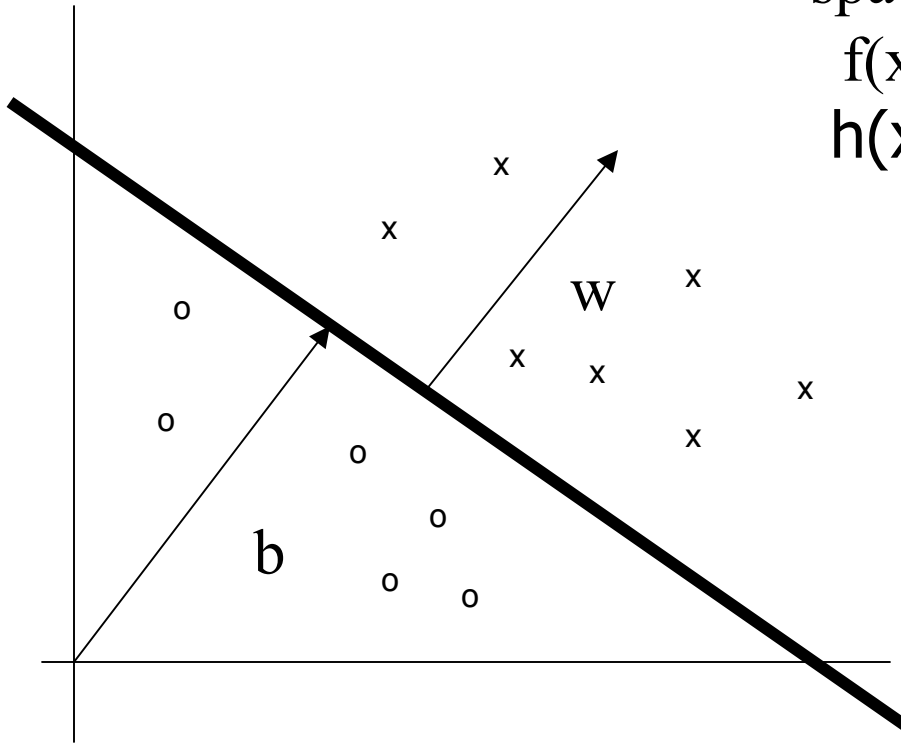
- Simplest case: classification using hyperplane in input space
- The Perceptron Algorithm (Rosenblatt, 57)
- Notation:
 - input space: $x \in X$
 - output space: $y \in Y = \{-1, +1\}$
 - hypothesis: $h \in H$
 - function: $f: X \rightarrow \mathbb{R}$
 - training Set: $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$
 - test error: ε
 - dot product $\langle x, z \rangle$

Perceptron

Linear separation of the input space

$$f(x) = \langle w, x \rangle + b$$

$$h(x) = \text{sign}(f(x))$$



Update rule:

if $y_i \langle w_k, x_i \rangle + b \leq 0$ then

$$w_{k+1} = w_k + \eta y_i x_i$$

$k++$

Observation

- Solution is linear combination of training points:

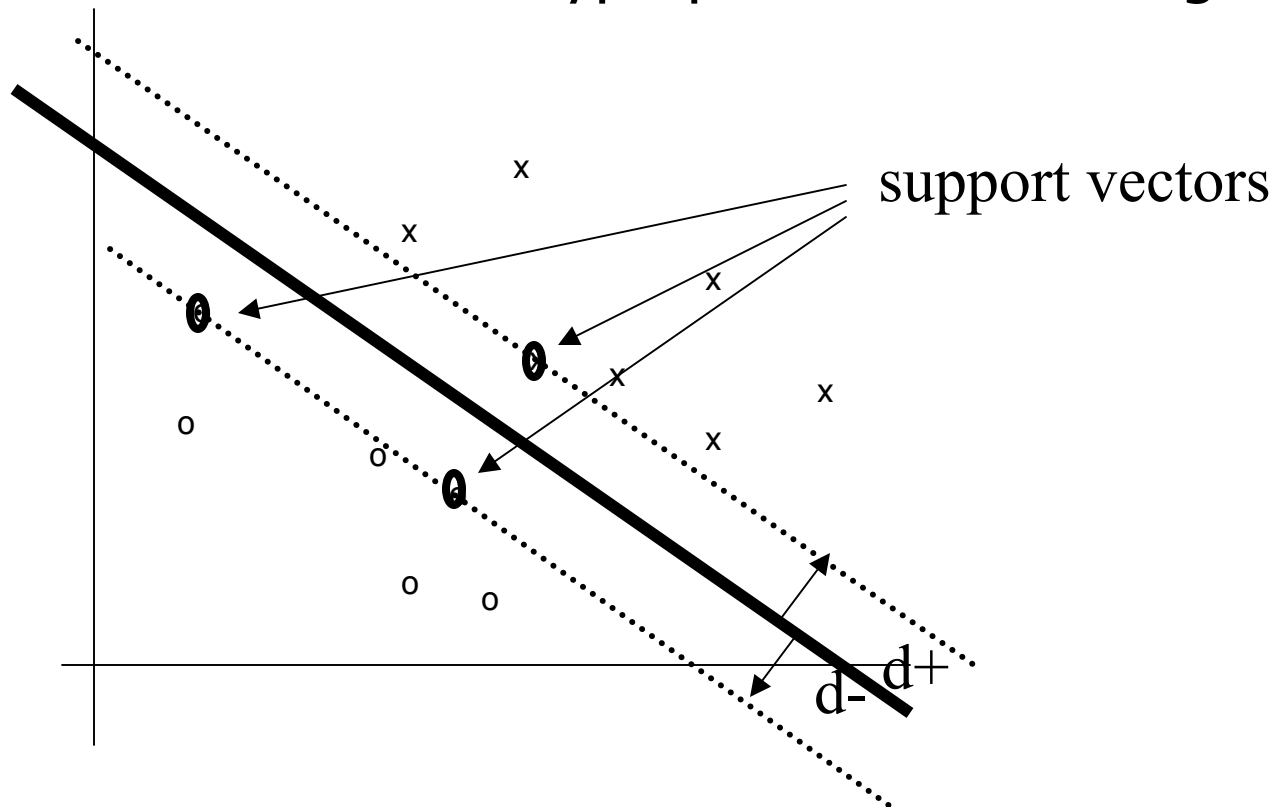
$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i, \quad \alpha_i \geq 0$$

- Only used informative points (mistake driven)
- can represent in dual form:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \sum \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b$$

Linear Support Vector Machines

- d_+ distance to closest positive example, d_- distance to closest negative example.
- Define margin of separating hyperplane to be $d_+ + d_-$
- SVMs look for hyperplane with the largest margin



Constrained Optimization

$$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} \geq +1 \text{ for } y_i = +1 \quad (1)$$

$$\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} \leq -1 \text{ for } y_i = -1 \quad (2)$$

combine: $y_i(\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) - 1 \geq 0 \quad \forall i$

for points for which equality (1) hold, lie on hyperplane H_1 :

$$H_1 : \mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} = 1$$

with normal $\mathbf{w}$ and perpendicular distance from the origin $\frac{|1 - \mathbf{b}|}{\|\mathbf{w}\|}$

for points for which equality (2) hold, lie on hyperplane H_2 :

$$H_2 : \mathbf{x}_i \cdot \mathbf{w} + \mathbf{b} = -1$$

with normal $\mathbf{w}$ and perpendicular distance from the origin $\frac{|-1 - \mathbf{b}|}{\|\mathbf{w}\|}$

$$d_+ = d_- = \frac{1}{\|\mathbf{w}\|}$$

margin $\frac{2}{\|\mathbf{w}\|}$; can maximize margin by minimizing $\|\mathbf{w}\|^2$ subject to constraints

Lagrangian

- Introduce Lagrange multipliers α_I

$$L_P \equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^I \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^I \alpha_i$$

- Minimize L_P with respect to w , b and $\alpha_I \geq 0$
- convex programming problem

- can equivalently solve the dual problem L_D

$$L_D = \sum_{i=1}^I \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

- maximize L_D with respect to constraints on w , b and $\alpha_I \geq 0$

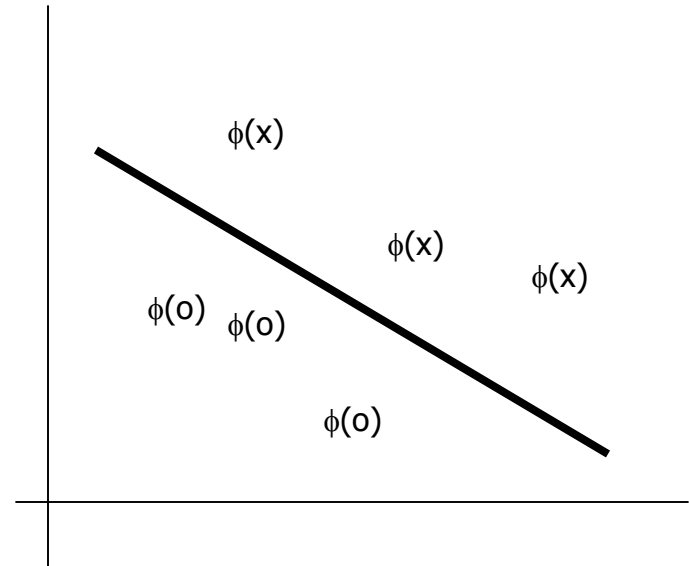
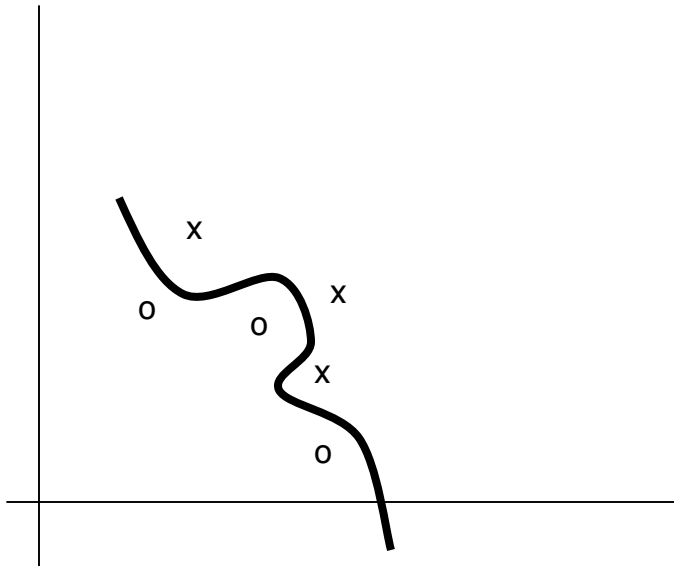
Limitations of LSVMs

- Linear classifiers cannot deal with
 - nonlinearly separable data
 - noisy data
- One solution: network of simple linear classifiers:
Neural Network
- Other solution: map data into a richer feature space which includes non-linear features, then use a linear classifier

SVMs cont.

- Map data into feature space where they are linearly separable

$$x \rightarrow \phi(x)$$



SVMs cont.

- In dual representation, the data points only appear inside dot products:

$$f(\mathbf{x}) = \sum \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle + b$$

- Introduce Kernels
- function that returns the value of the dot product between the images of the two arguments

$$K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle$$

- solves computational problem of working in high dimensional space

SVMs summary

- Properties: duality, kernels, margin, convexity
- based on statistical learning theory; connections to VC-dimension
- very general and rich class of pattern recognition methods

Evaluating and Comparing Classifiers

- estimating true future error rate
- reclassify the training data and see the proportion that is misclassified. Called the *apparent* or *resubstitution* error rate. This is typically **underestimate** of future proportion and is ill-advised.
- Estimate future error by calculating error on new sample, the **test set**
- If data set is small, use cross-validation: leave some proportion of the data and evaluate on
 - leave-one-out

Cross-validation

- If data set is small, use cross-validation: leave out some proportion of the data when classifier is constructed and evaluate on left out portion. Repeat, with different parts of the data being omitted.
 - leave-one-out: leave out one point at a time
 - bootstrap methods:
 - Introduced by Efron, 1979, to calculate confidence intervals for parameters when sample size is small
 - at a high-level:
 - Observations randomly re-assigned and estimates are recomputed; treated as repeated experiments

Feature Selection

- classification in high dimensions--Too many variables!
- Not all variables may be necessary for accurate discrimination
- variable selection:
- select subset p' of original p variables.
 - part of algorithm, e.g. tree inducer
 - wrapper: external loop that systematically adds and removes variables from the current set
- variable transformation
 - pca, etc.
 - disadvantage: not necessarily matched for optimizing classification performance

References

- *Principles of Data Mining*, Hand, Mannila, Smyth. MIT Press, 2001.
- *Data Mining*, Witten and Eibe. Academic Press, 2000.
- *Machine Learning*, T. Mitchell. McGraw-Hill, 1997.
- Nello Cristianini's ICML 2001 tutorial,
<http://www.support-vector.net/icml-tutorial.pdf>
- Cristianini, N., Shawe-Taylor, J., [*An Introduction to Support Vector Machines*](#), Cambridge University Press, (2000).
- Schölkopf, S., Burges, C. J. C., Smola, A. J., [*Advances in Kernel Methods: Support Vector Learning*](#), MIT Press, Cambridge, MA, (1999).