Non-Parametric Models

Review of last class:
Decision Tree Learning

- dealing with the overlearning problem: pruning
- ensemble learning
- boosting

Agenda

- Nearest neighbor models
- Finding nearest neighbors with kd trees
- Locality-sensitive hashing
- Nonparametric regression

Non-Parametric Models

- doesn’t mean that the model lacks parameters
- parameters are not known or fixed in advance
- make no assumptions about probability distributions
- instead, structure determined from the data

Comparison of Models

Parametric

- data summarized by a fixed set of parameters
- once learned, the original data can be discarded
- good when data set is relatively small – avoids overfitting
- best when correct parameters are chosen!

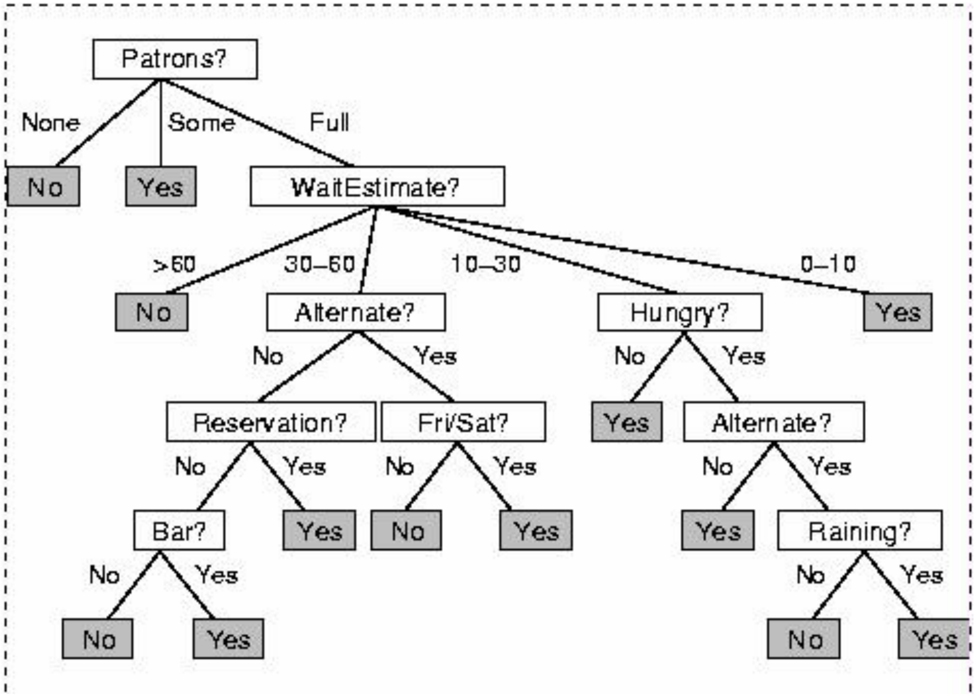
Non-Parametric

- data summarized by an unknown (or non-fixed) set of parameters
- must keep original data to make predictions or to update model
- may be slower, but generally more accurate

Instance-Based Learning

Decision Trees

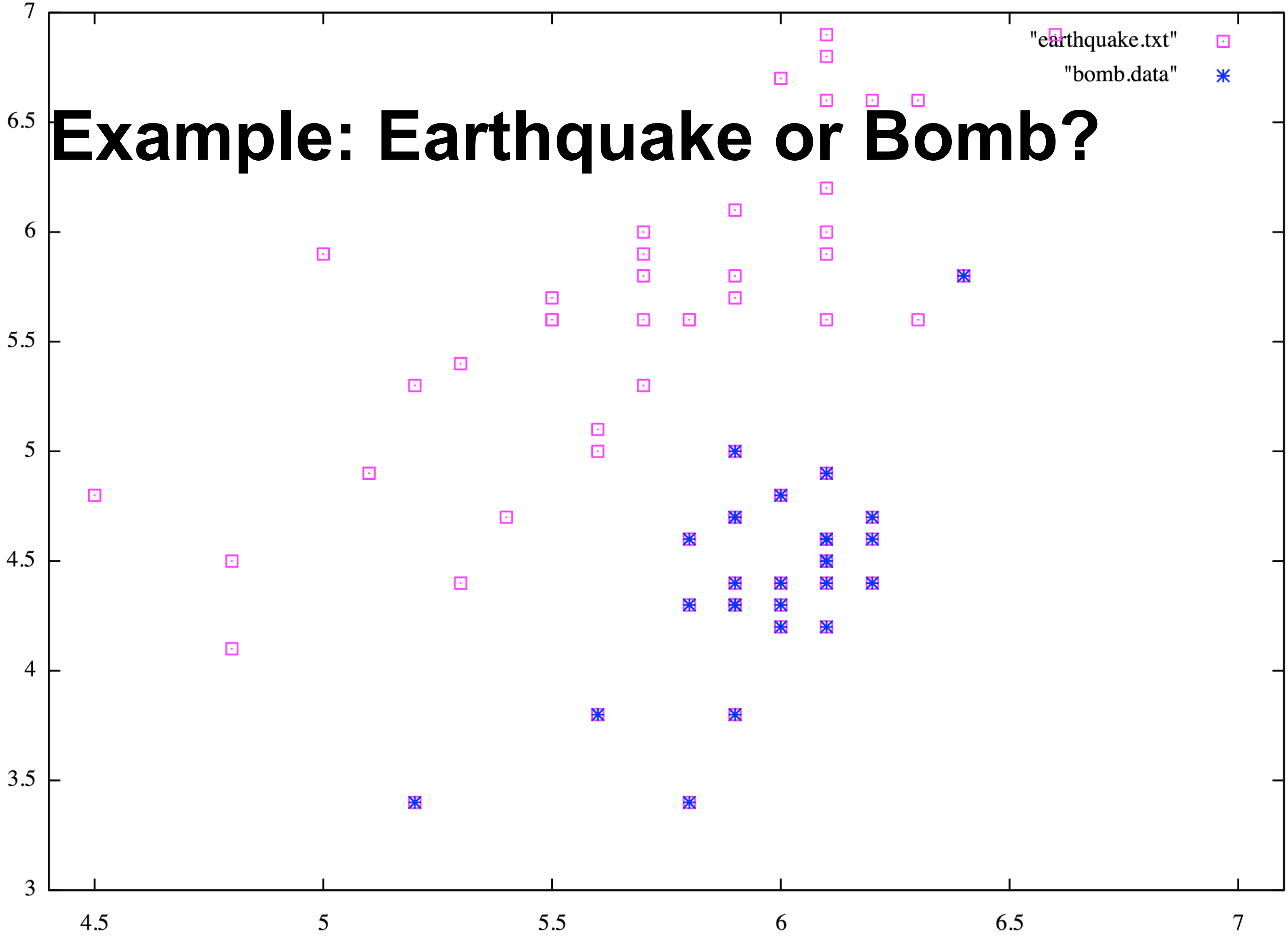
- examples (training set) described by:
  - input: the values of attributes
  - output: the classification (yes/no)
- can represent any Boolean function



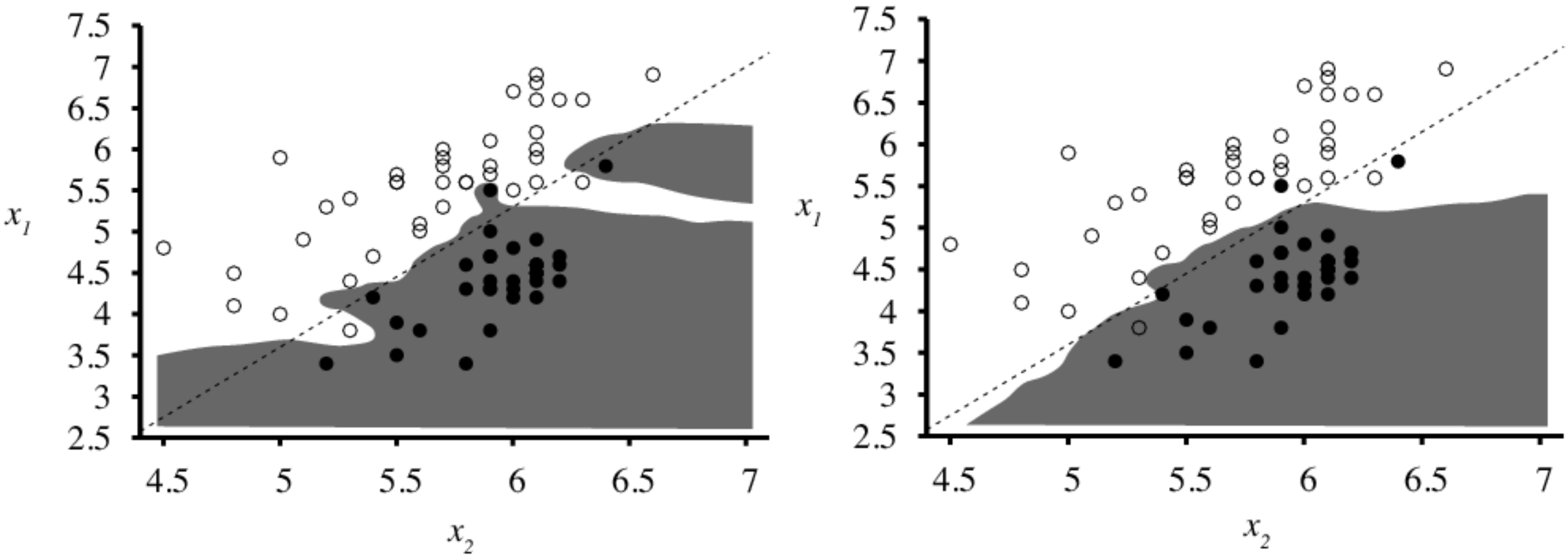
Another NPM approach: Nearest neighbor (k-NN) models

- given query $\mathbf{x}_q$
- answer query by finding the *k* examples nearest to $\mathbf{x}_q$
- classification:
  - take plurality vote (majority for binary classification) of neighbors
- regression
  - take mean or median of neighbor values

Example: Earthquake or Bomb?



Modeling the data with k-NN



k = 1

k = 5

Measuring "nearest"

- Minkowski distance calculated over each attribute (or dimension) *i*

$$L^p(\mathbf{x_j}, \mathbf{x_q}) = \left(\sum_i | x_{j,i} - x_{q,i} |^p\right)^{1/p}$$

  - p = 2: Euclidean distance – typically used if dimensions measure similar properties (e.g., width, height, depth)
  - p = 1: Manhattan distance – if dimensions measure dissimilar properties (e.g., age, weight, gender)

Recall a problem we faced before

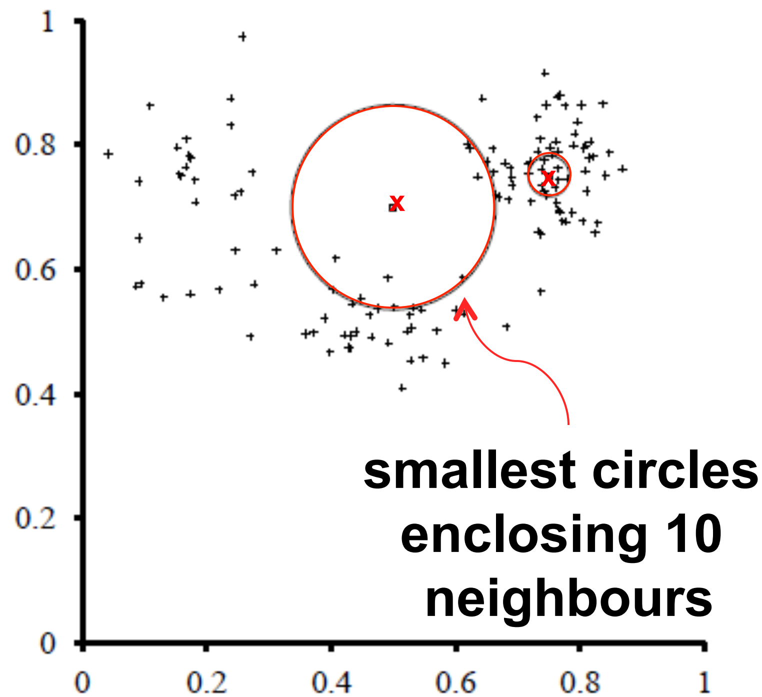
- shape of the data looks very different depending on the scale
  - e.g., height vs. weight, with height in mm or km
- similarly, with k-NN, if we change the scale, we'll end up with different neighbors

Simple solution

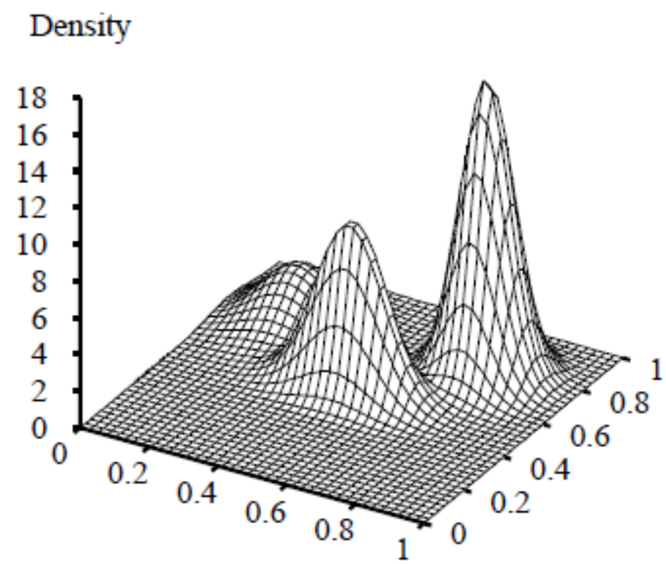
- simple solution is to normalize:

$$x'_{j,i} = (x_{j,i} - \mu_i) / \sigma_i$$

Example: Density estimation



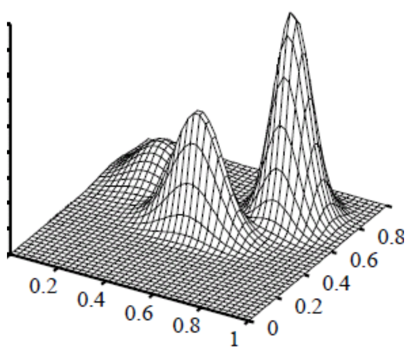
128-point sample



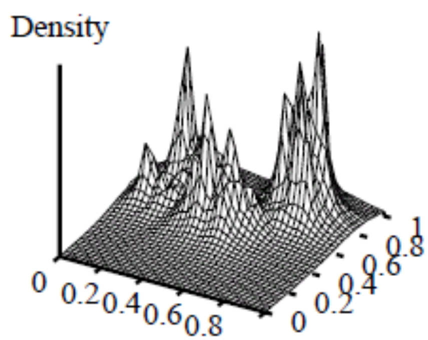
MoG representation

Density Estimation using k-NN

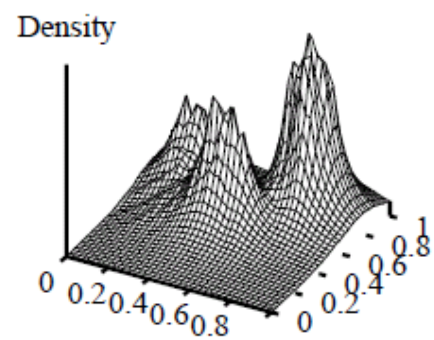
- # of neighbours impacts quality of estimation



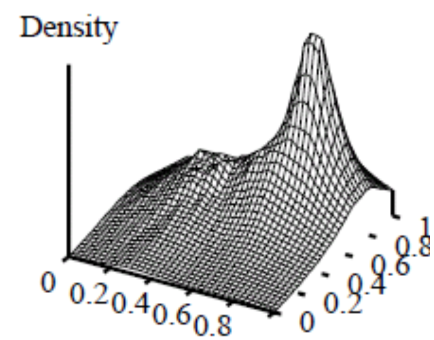
ground truth



k=3



k=10



k=40

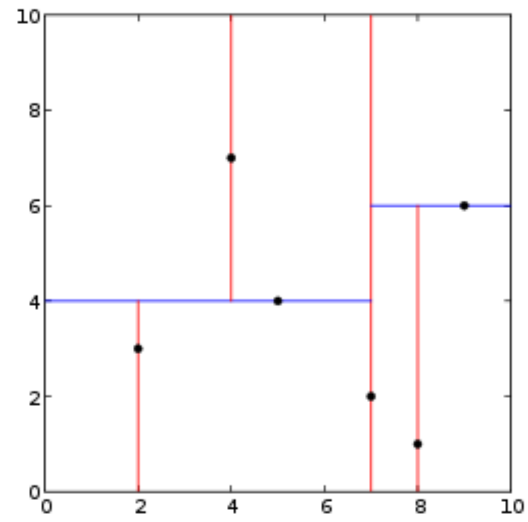
Curse of dimensionality

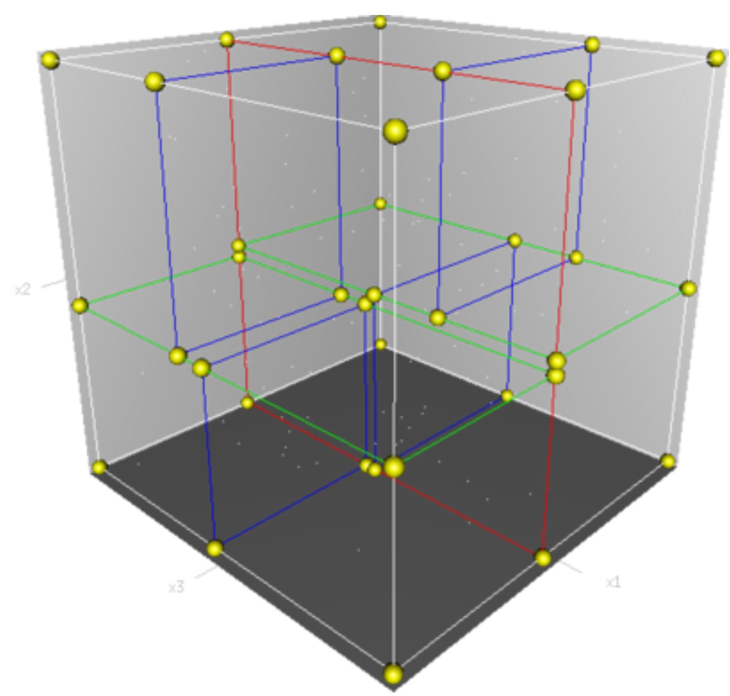
- we want to find k = 10 nearest neighbors among N=1,000,000 points of an n-dimensional space
- sounds easy, right?
- volume of neighborhood is k/N
- average side length l of neighborhood is $(k/N)^{1/n}$

n	l
1	.00001
2	.003
3	.002
10	.3
20	.56

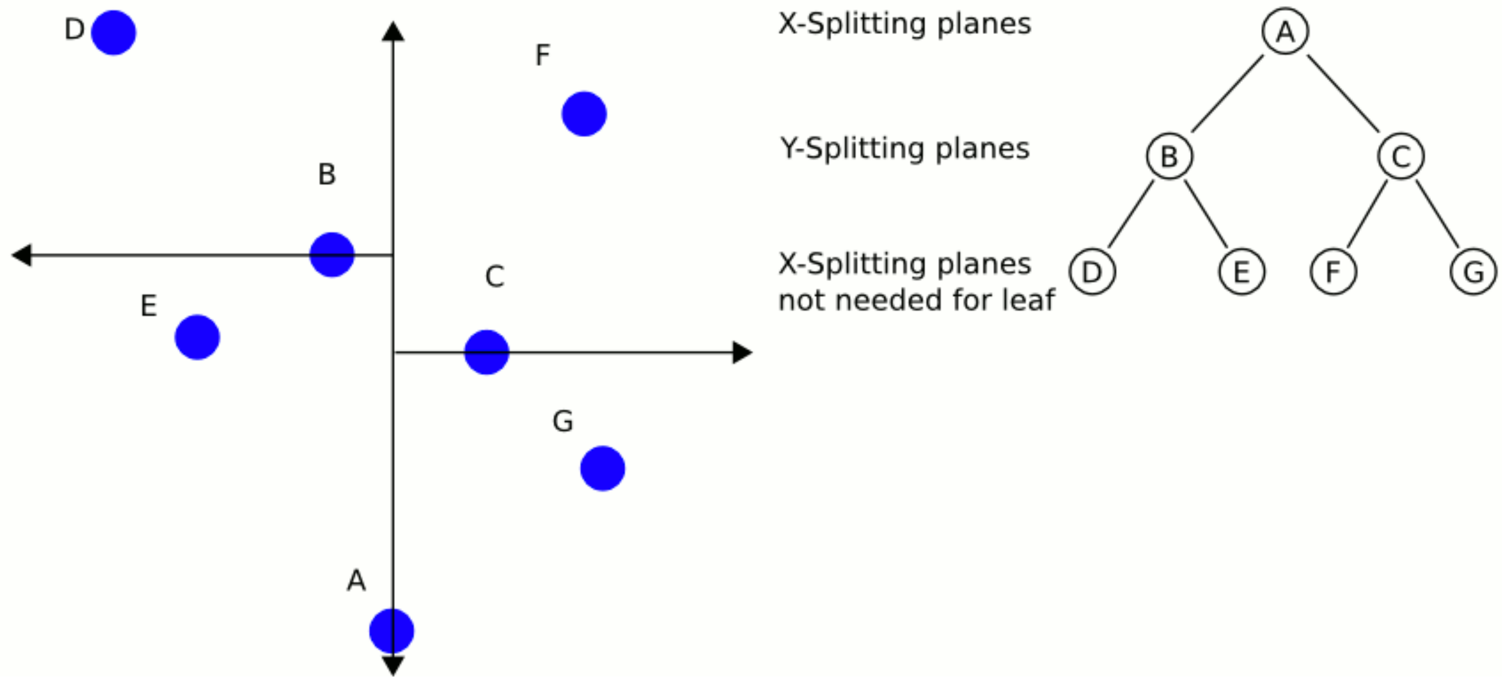
k-dimensional (kd) trees

- balanced binary tree with arbitrary # of dimensions
- data structure that allows efficient lookup of nearest neighbors (when # of examples >> k)
- recursively divides data into left and right branches based on value of dimension *i*





k-dimensional (kd) trees



- query value might be on left half of divide but have some of *k* nearest neighbors on right half
- decide whether to inspect the right half based on distance of best match found from dividing hyperplane

Locality-Sensitive Hashing (LSH)

- uses a combination of n random projections, built from subsets of the bit-string representation of each value
- value of each of the n projections stored in the associated hash bucket

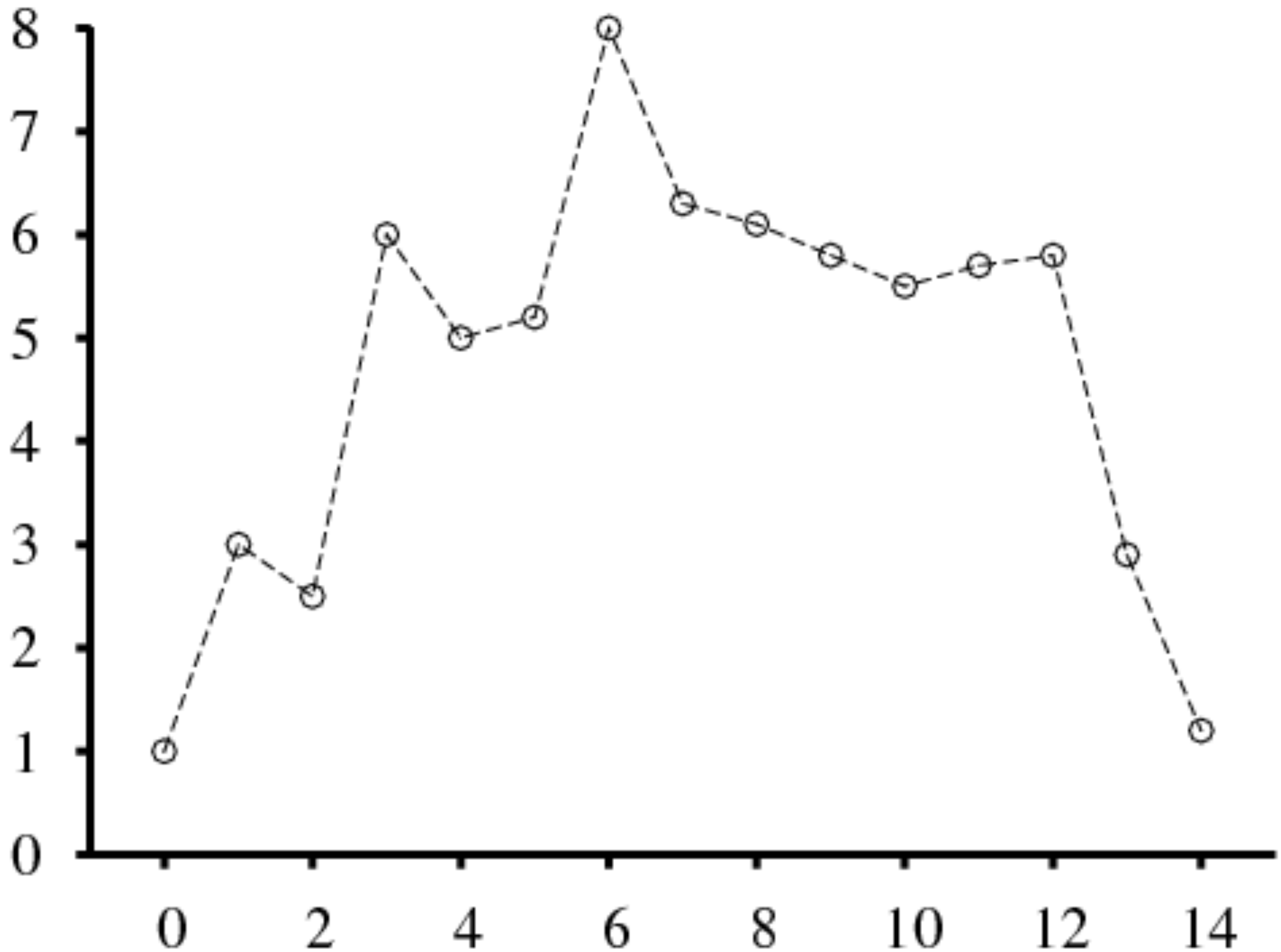
Locality-Sensitive Hashing (LSH)

- on search, the set of points from all hash buckets corresponding to the query are combined together
- then measure distance from query value to each of the returned values
- real-world example:
  - data set of 13 million samples of 512 dimensions
  - LSH only needs to examine a few thousand images
  - 1000-fold improvement over kd trees!

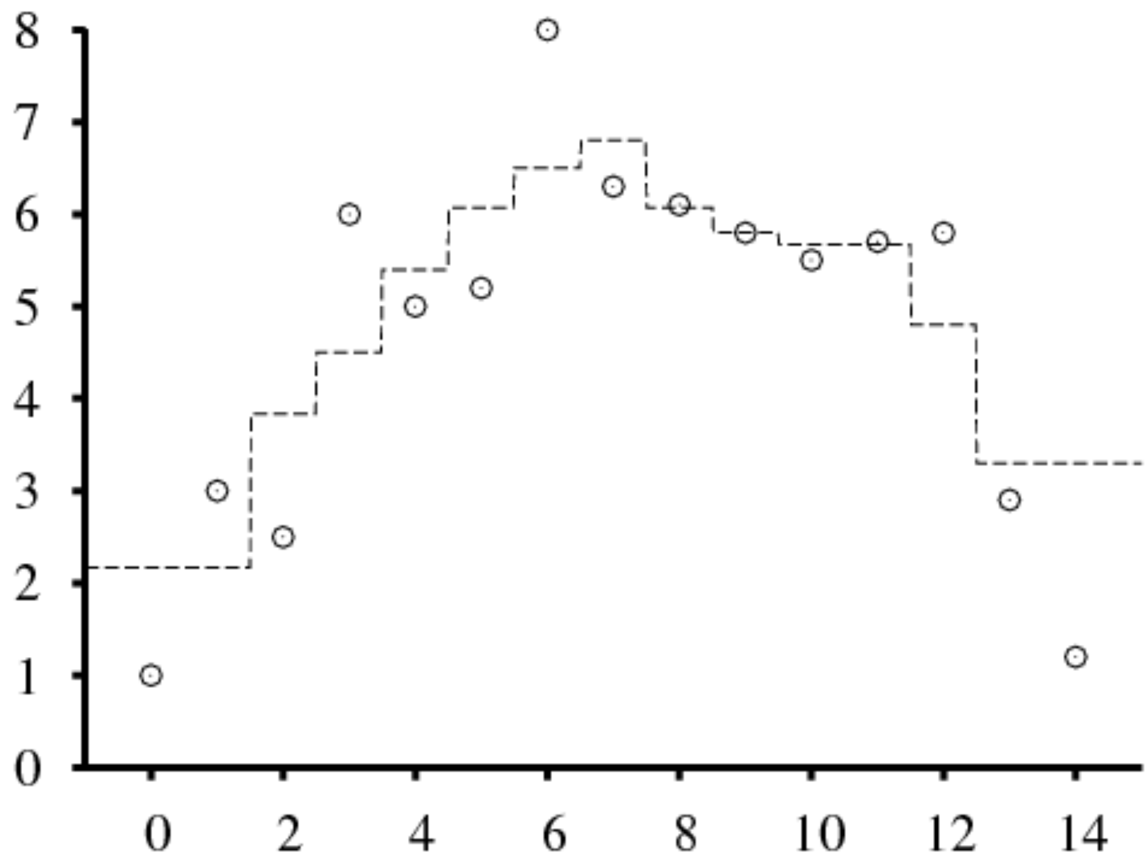
Nonparametric Regression Models

- Let’s see how different NPM strategies fare on a regression problem

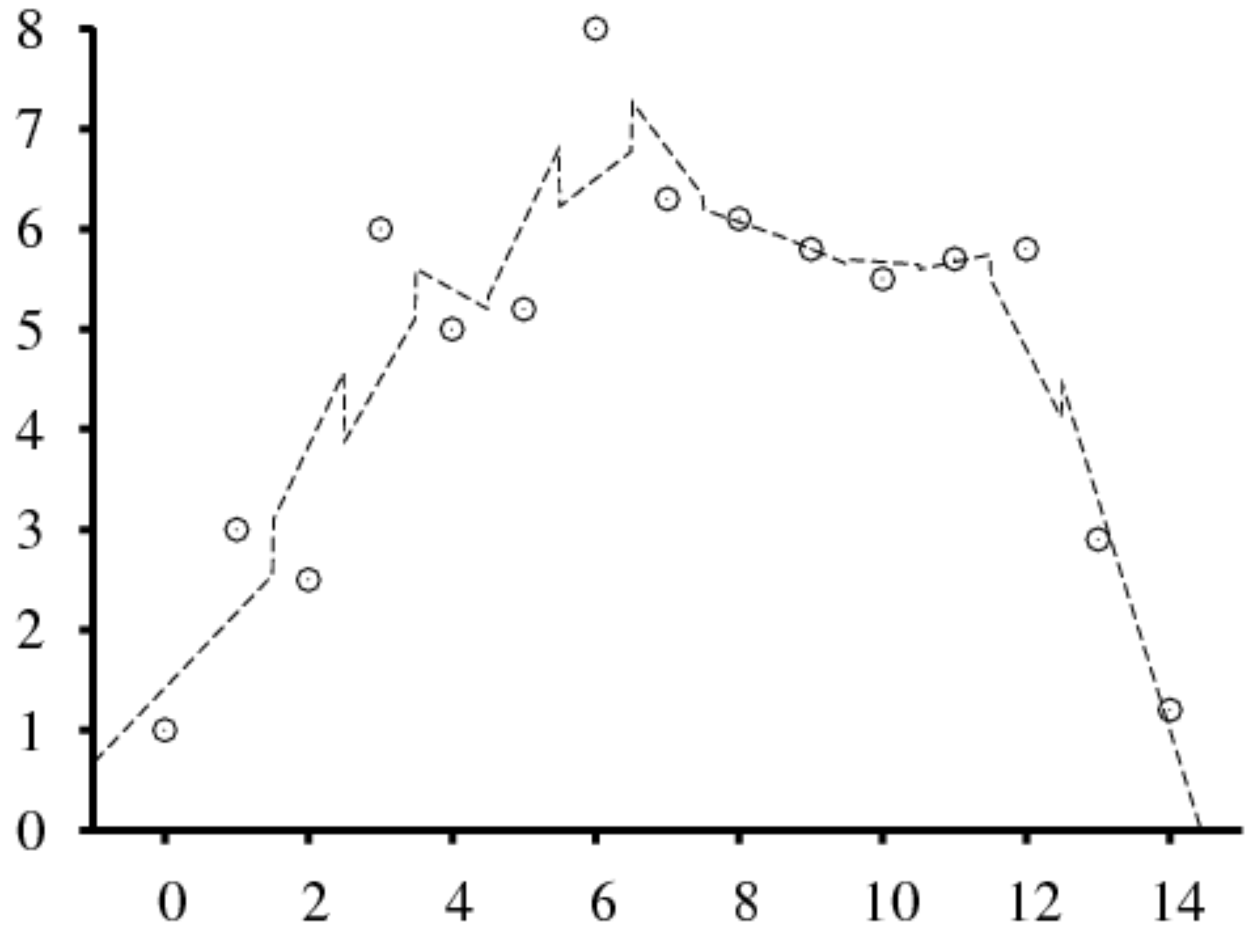
Piecewise linear regression



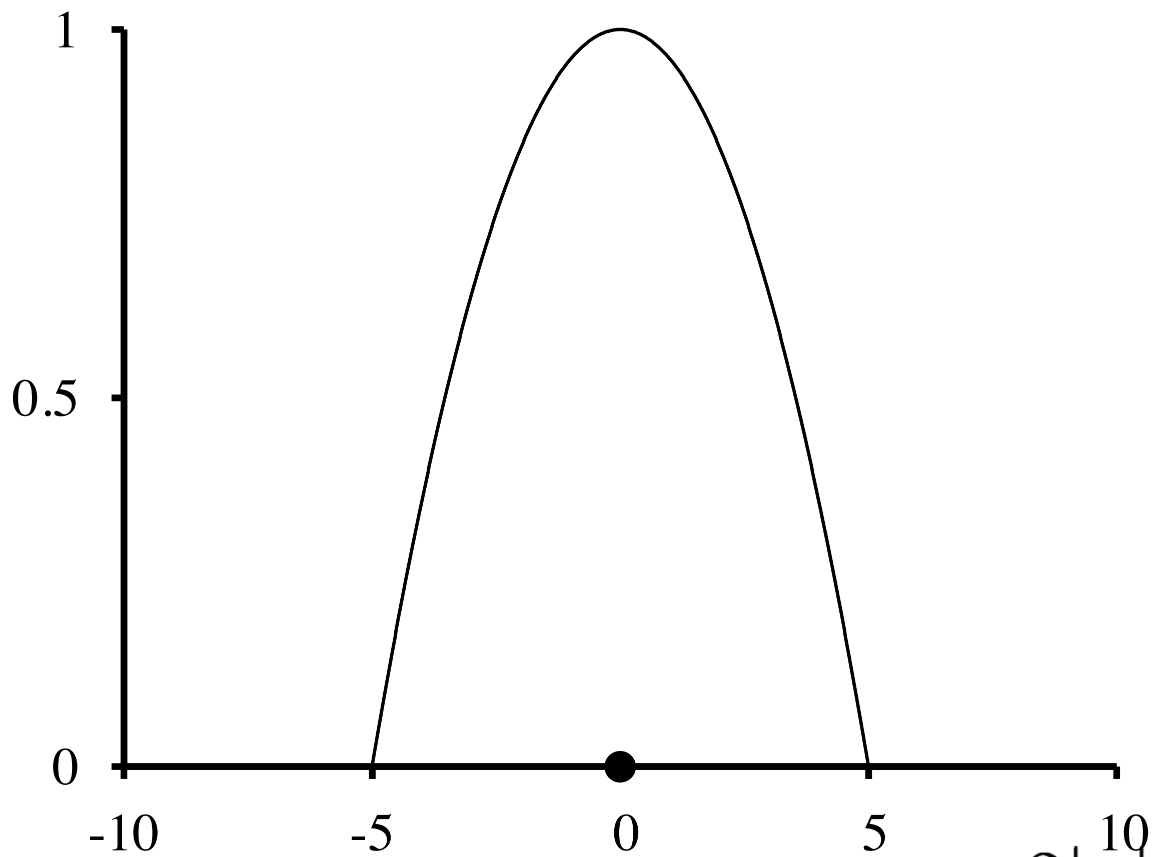
3-NN Average



Linear regression through 3-NN



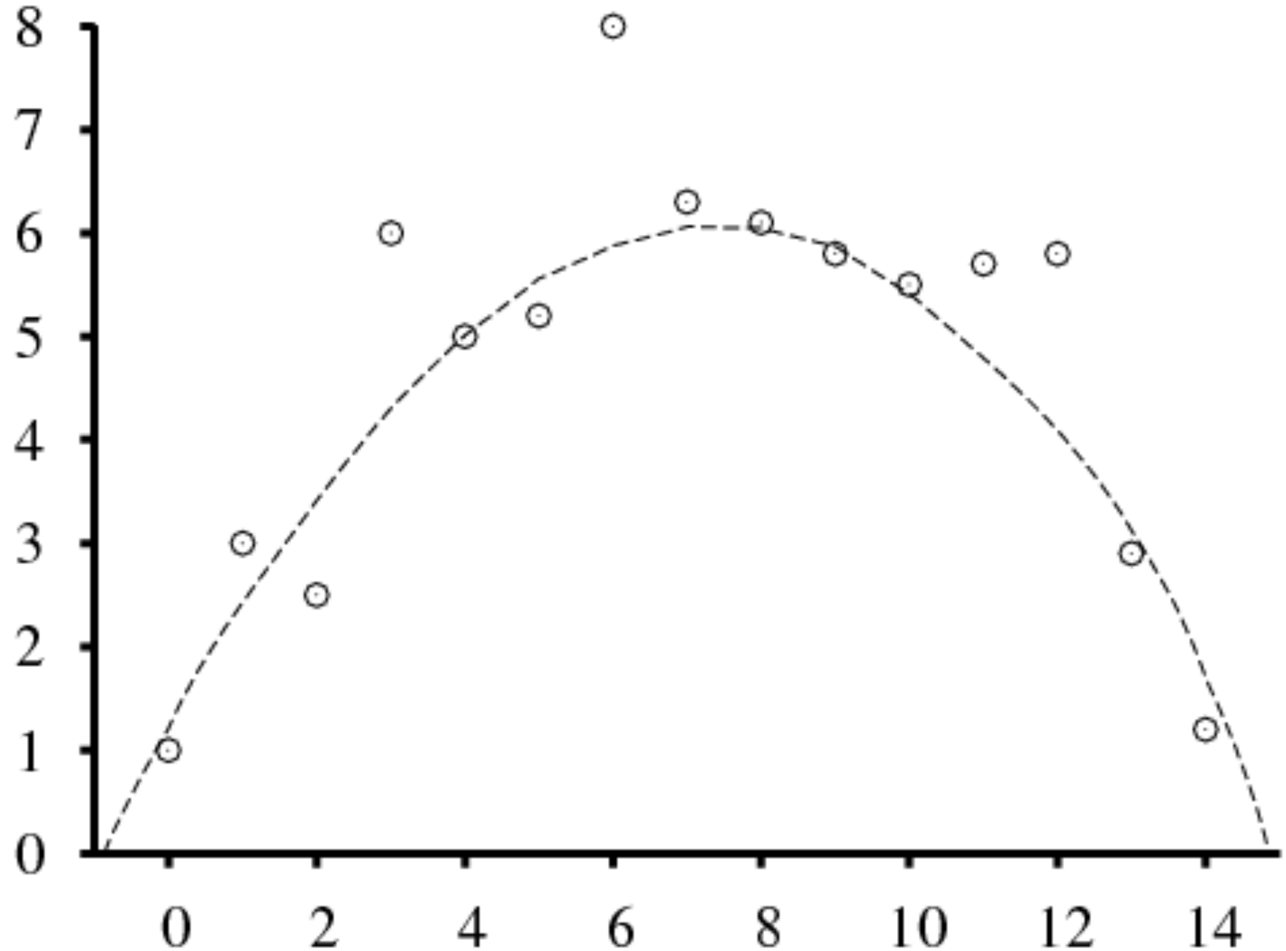
Local weighting of data with kernel



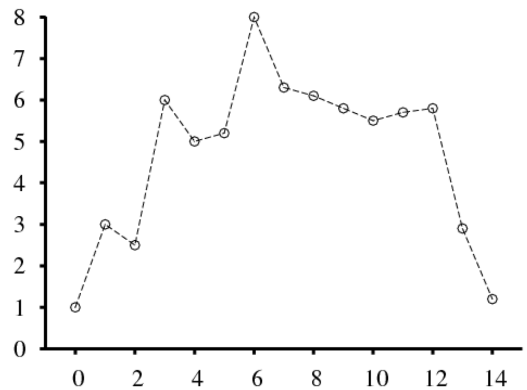
quadratic kernel with k = 10:

$$w = \max(0, 1 - (\frac{2|x|}{k})^2)$$

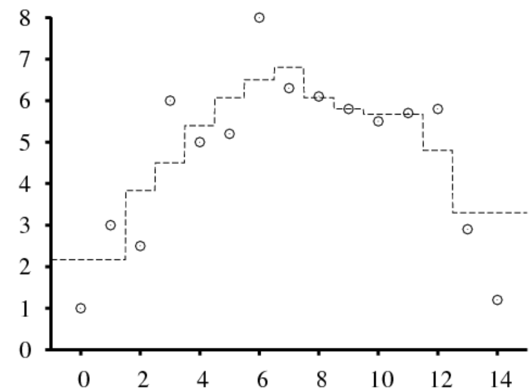
Locally weighted quadratic kernel k=10



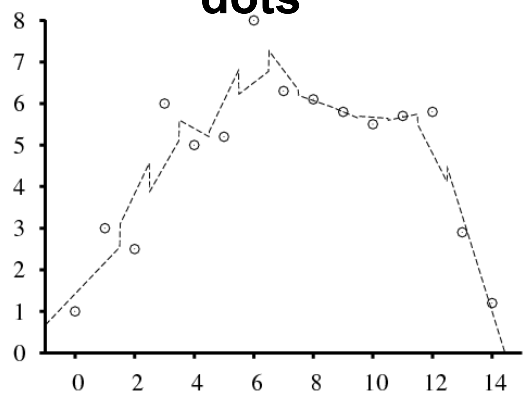
Comparison



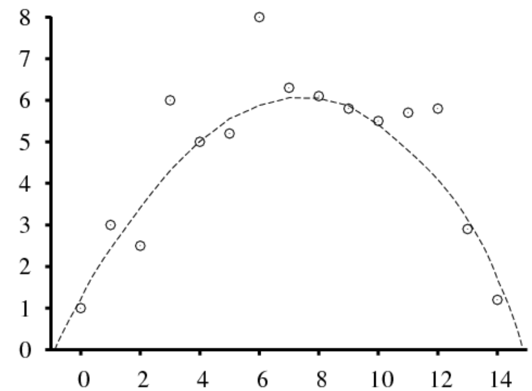
connect the dots



3-NN average



3-NN linear regression



locally weighted regression (quadratic kernel width k=10)

Next class

- Statistical learning methods, Ch. 20