ECE 5424: Introduction to Machine Learning

Topics:

- (Finish) Nearest Neighbor

Readings: Barber 14 (kNN)

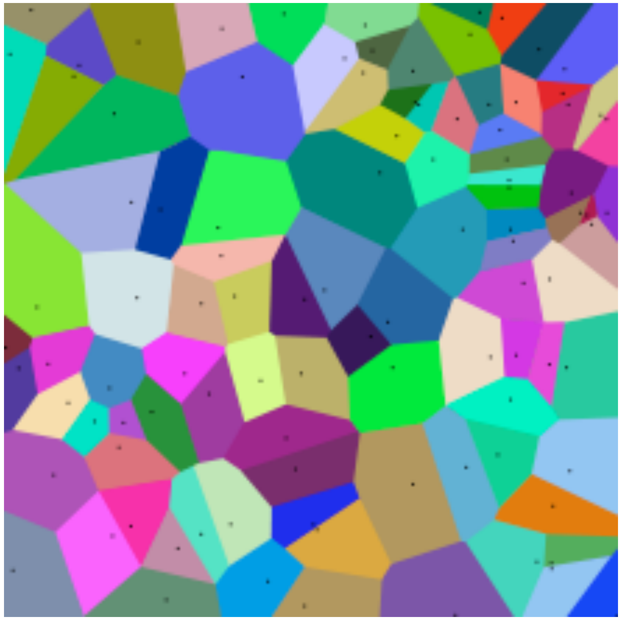
Stefan Lee
Virginia Tech

Administrative

- HW1
  - Out now on Scholar
  - Due on Wednesday 09/14, 11:55pm
  - Please please please please please start early
  - Implement K-NN
  - Kaggle Competition
  - Bonus points for best performing entries.
  - Bonus points for beating the instructor/TA.

Recap from last time

Nearest Neighbours



Instance/Memory-based Learning

Four things make a memory based learner:

- *A distance metric*

- *How many nearby neighbors to look at?*

- *A weighting function (optional)*

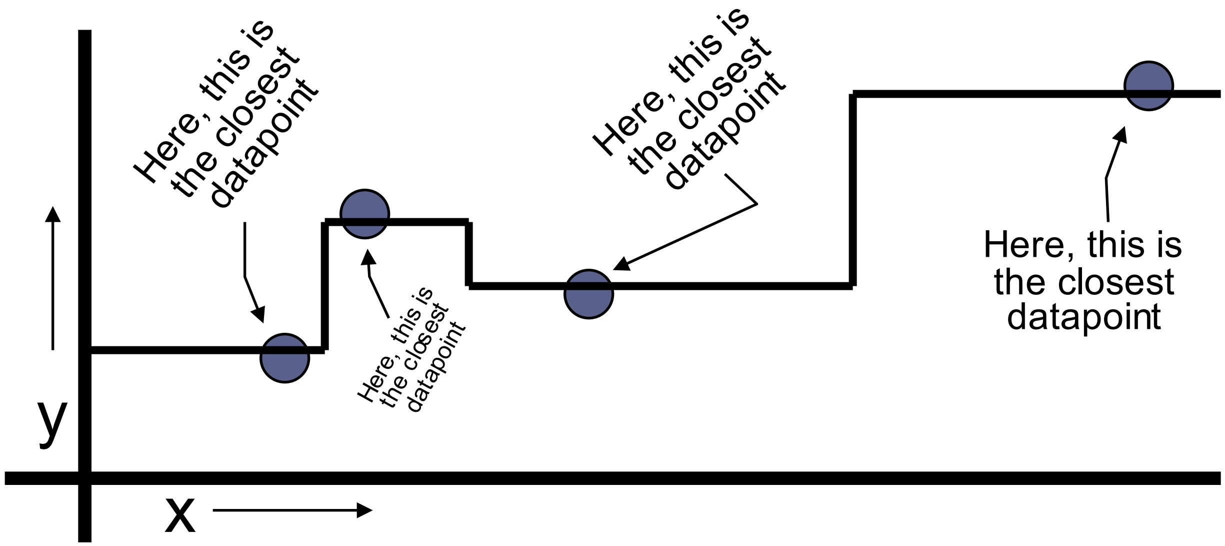
- *How to fit with the local points?*

1-Nearest Neighbour

Four things make a memory based learner:

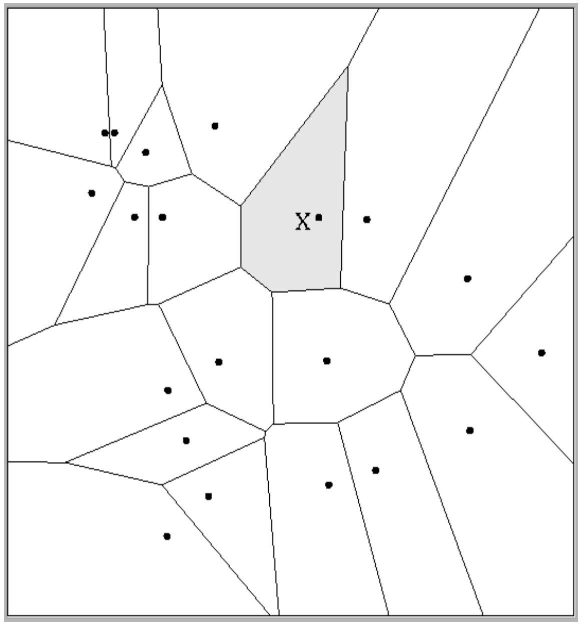
- *A distance metric*
  - **Euclidean (and others)**
- *How many nearby neighbors to look at?*
  - **1**
- *A weighting function (optional)*
  - **unused**
- *How to fit with the local points?*
  - **Just predict the same output as the nearest neighbour.**

1-NN for Regression

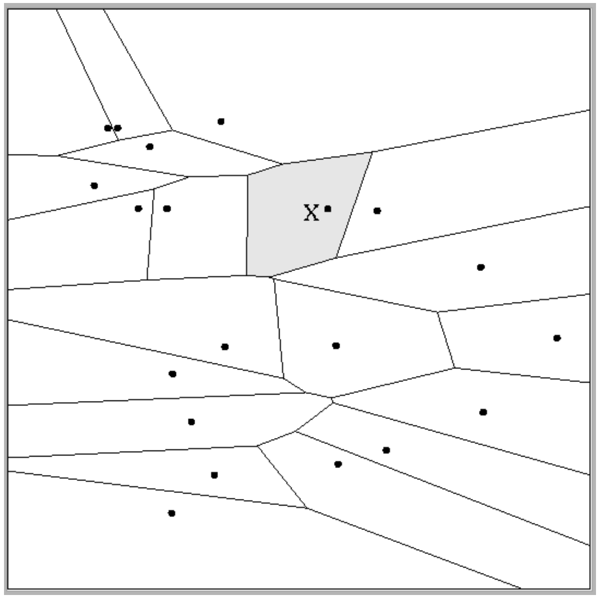


Multivariate distance metrics

Suppose the input vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$ are two dimensional:
$\mathbf{x}_1 = (x_{11} , x_{12}) , \mathbf{x}_2 = (x_{21} , x_{22}) , \ldots \mathbf{x}_N = (x_{N1} , x_{N2})$.
One can draw the nearest-neighbor regions in input space.



$Dist(\mathbf{x}_i,\mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2$



$Dist(\mathbf{x}_i,\mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (3x_{i2} - 3x_{j2})^2$

The relative scalings in the distance metric affect region shapes

Slide Credit: Carlos Guestrin

Euclidean distance metric

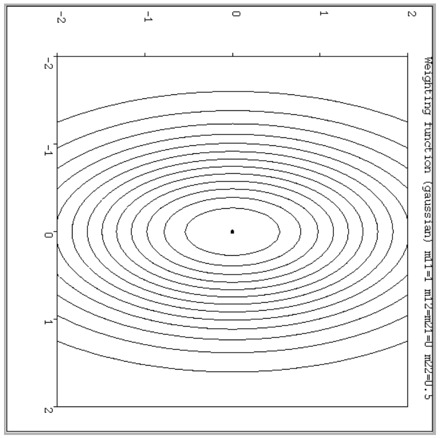
$$D(x, x') = \sqrt{\sum_i \sigma_i^2 (x_i - x'_i)^2}$$

Or equivalently,

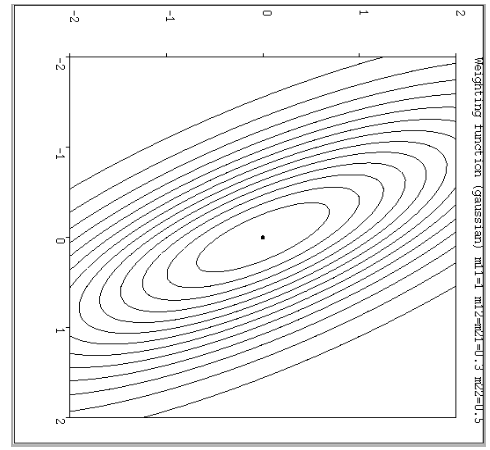
$$D(x, x') = \sqrt{(x_i - x'_i)^T A (x_i - x'_i)}$$

where $A = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{bmatrix}$

Notable distance metrics
(and their level sets)

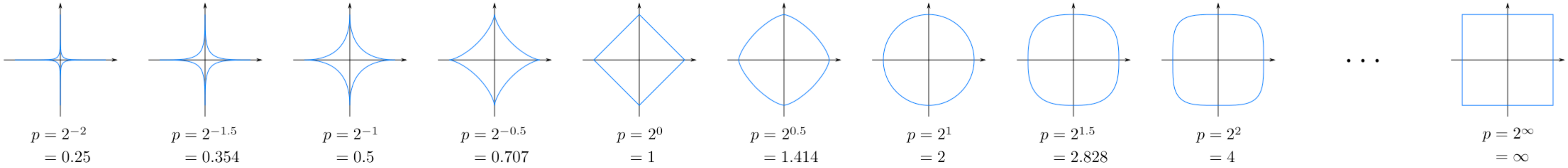


Scaled Euclidian (L_2)

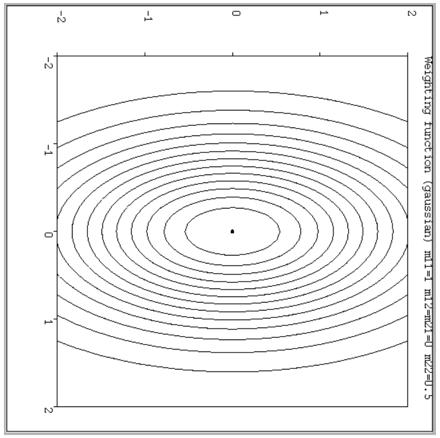


Mahalanobis
(non-diagonal A)

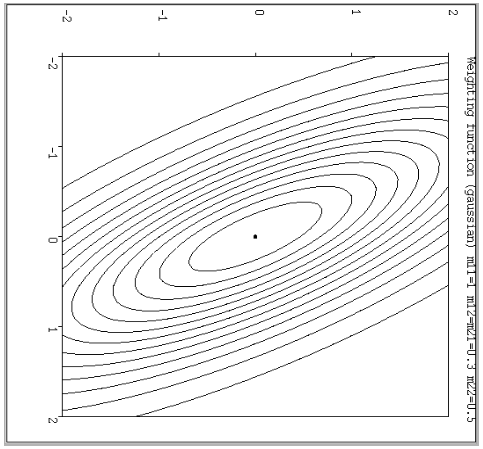
Minkowski distance



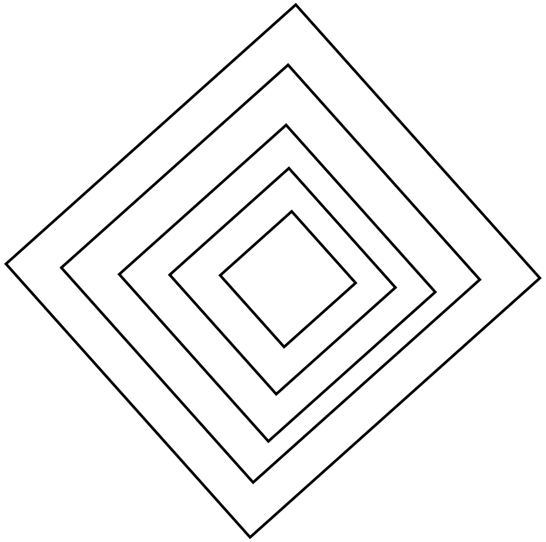
Notable distance metrics (and their level sets)



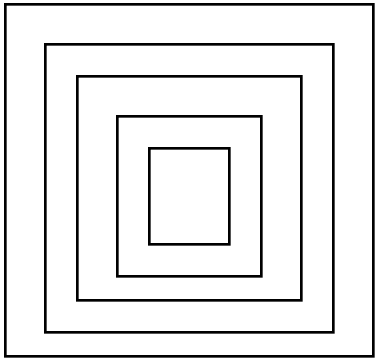
Scaled Euclidian (L_2)



Mahalanobis (non-diagonal A)



L_1 norm (absolute)



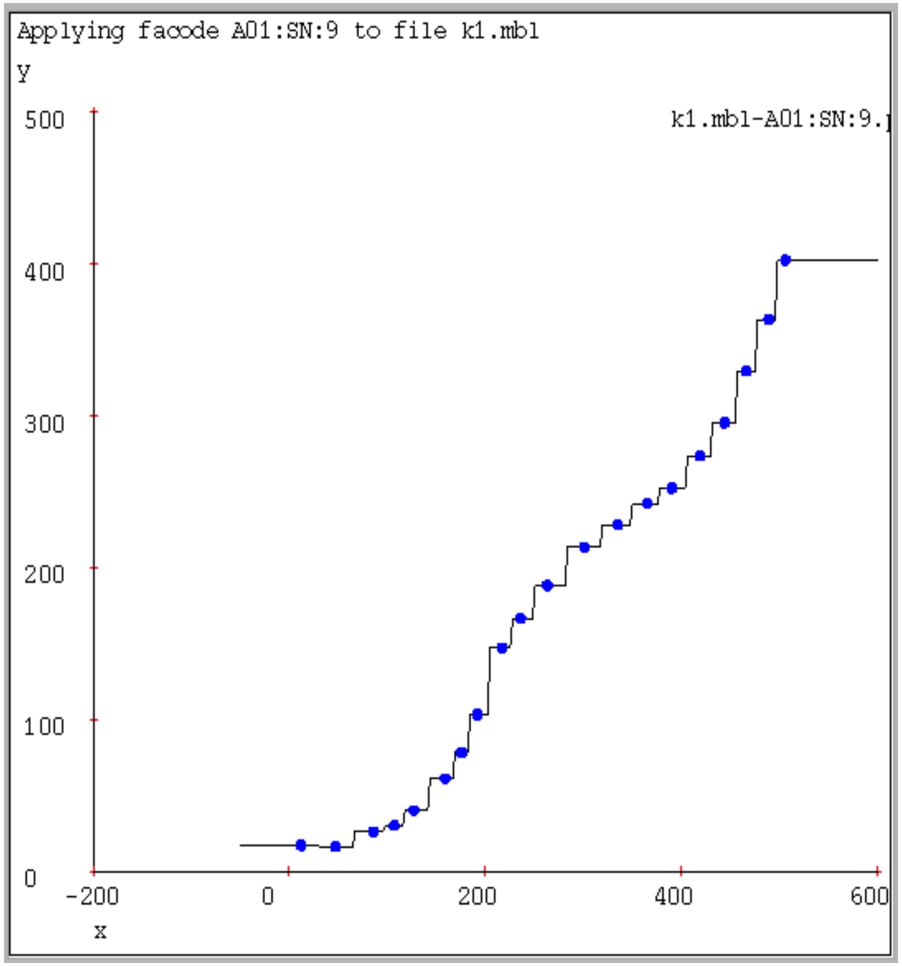
L_{inf} *(max) norm*

Plan for today

- (Finish) Nearest Neighbour
  - Kernel Classification/Regression
  - Curse of Dimensionality

1-NN for Regression

- Often bumpy (overfits)



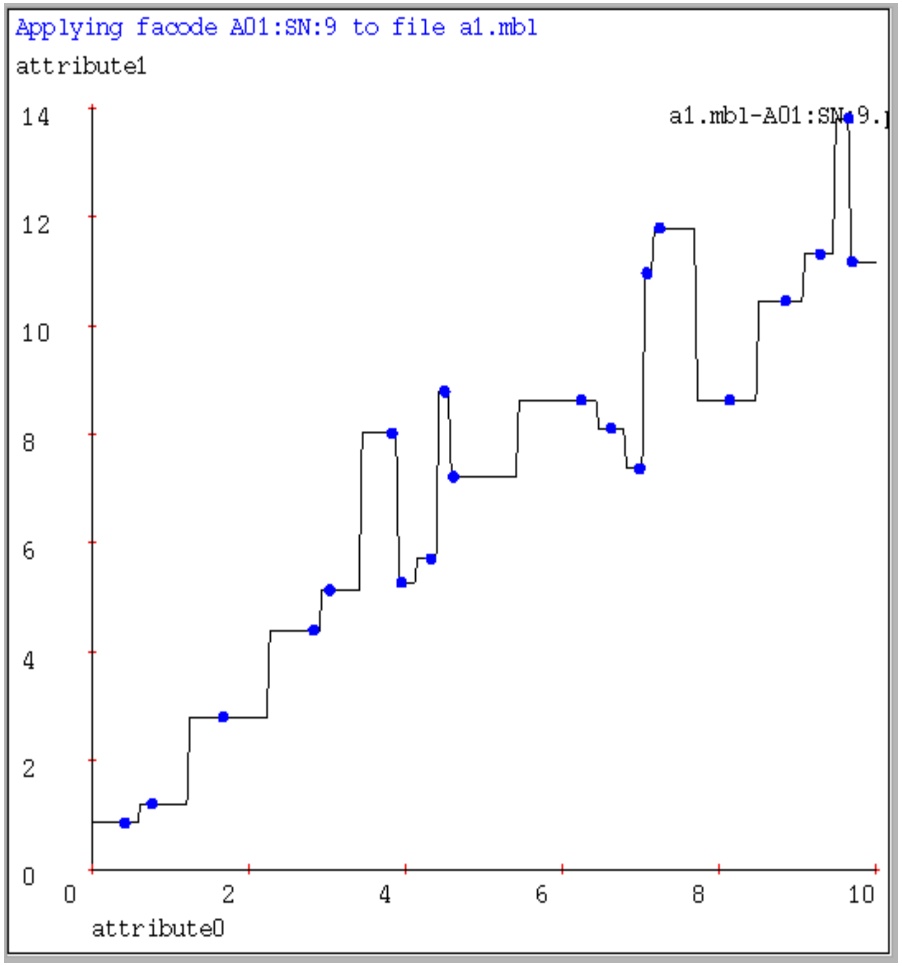
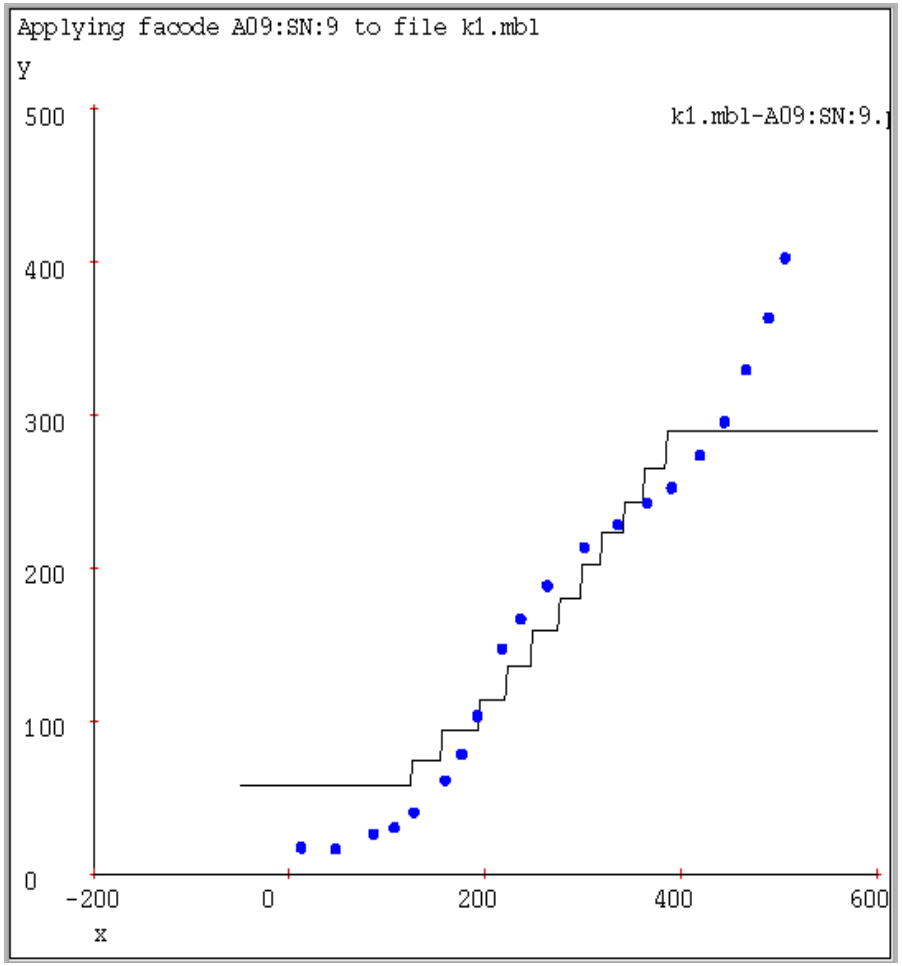


Figure Credit: Andrew Moore

9-NN for Regression

- Often bumpy (overfits)



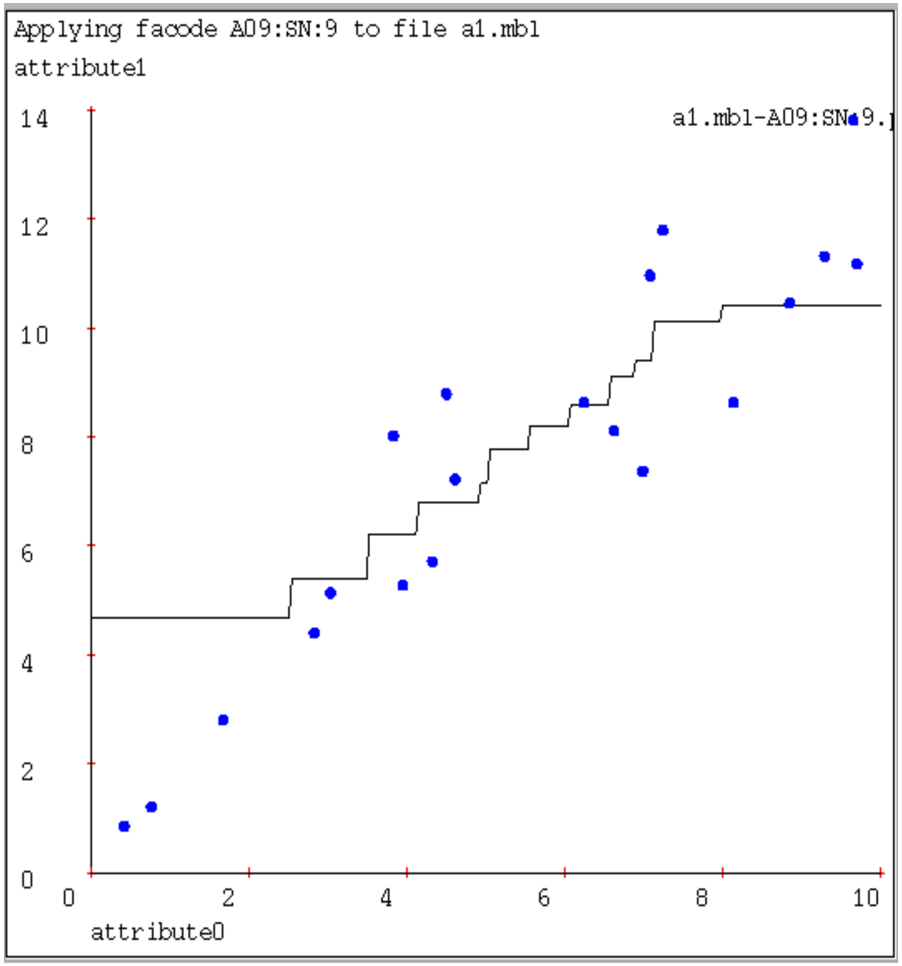


Figure Credit: Andrew Moore

Weighted k-NNs

- Neighbors are not all the same

Kernel Regression/Classification

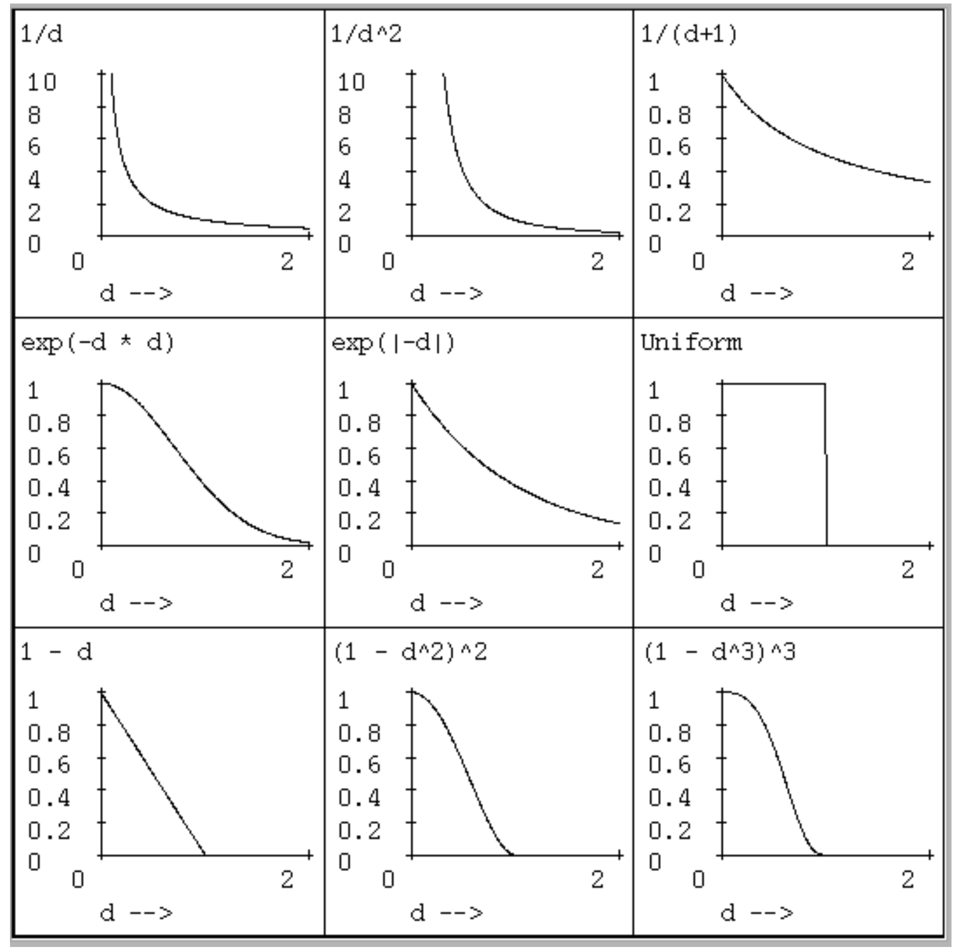
Four things make a memory based learner:

- *A distance metric*
  - **Euclidean (and others)**
- *How many nearby neighbors to look at?*
  - **All of them**
- *A weighting function (optional)*
  - $w_i = exp(-d(x_i, query)^2 / \sigma^2)$
  - Nearby points to the query are weighted strongly, far points weakly. The σ parameter is the **Kernel Width**. Very important.
- *How to fit with the local points?*
  - **Predict the weighted average of the outputs**

    predict = $\Sigma w_i y_i / \Sigma w_i$

Weighting/Kernel functions

$w_i = exp(-d(x_i, query)^2 / \sigma^2)$

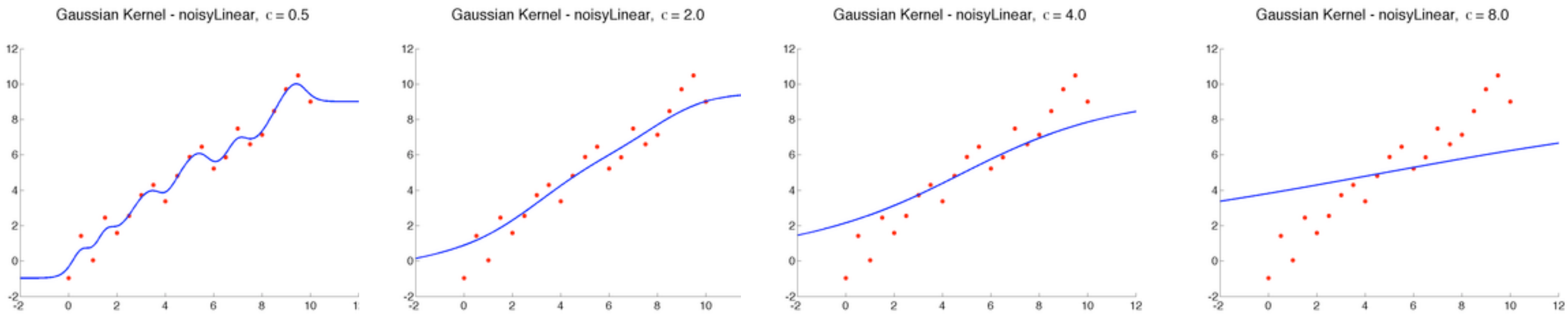


(Our examples use Gaussian)

Effect of Kernel Width

- What happens as σ→inf?

- What happens as σ→0?



Problems with Instance-Based Learning

- Expensive
  - No Learning: most real work done during testing
  - For every test sample, must search through all dataset – very slow!
  - Must use tricks like approximate nearest neighbour search

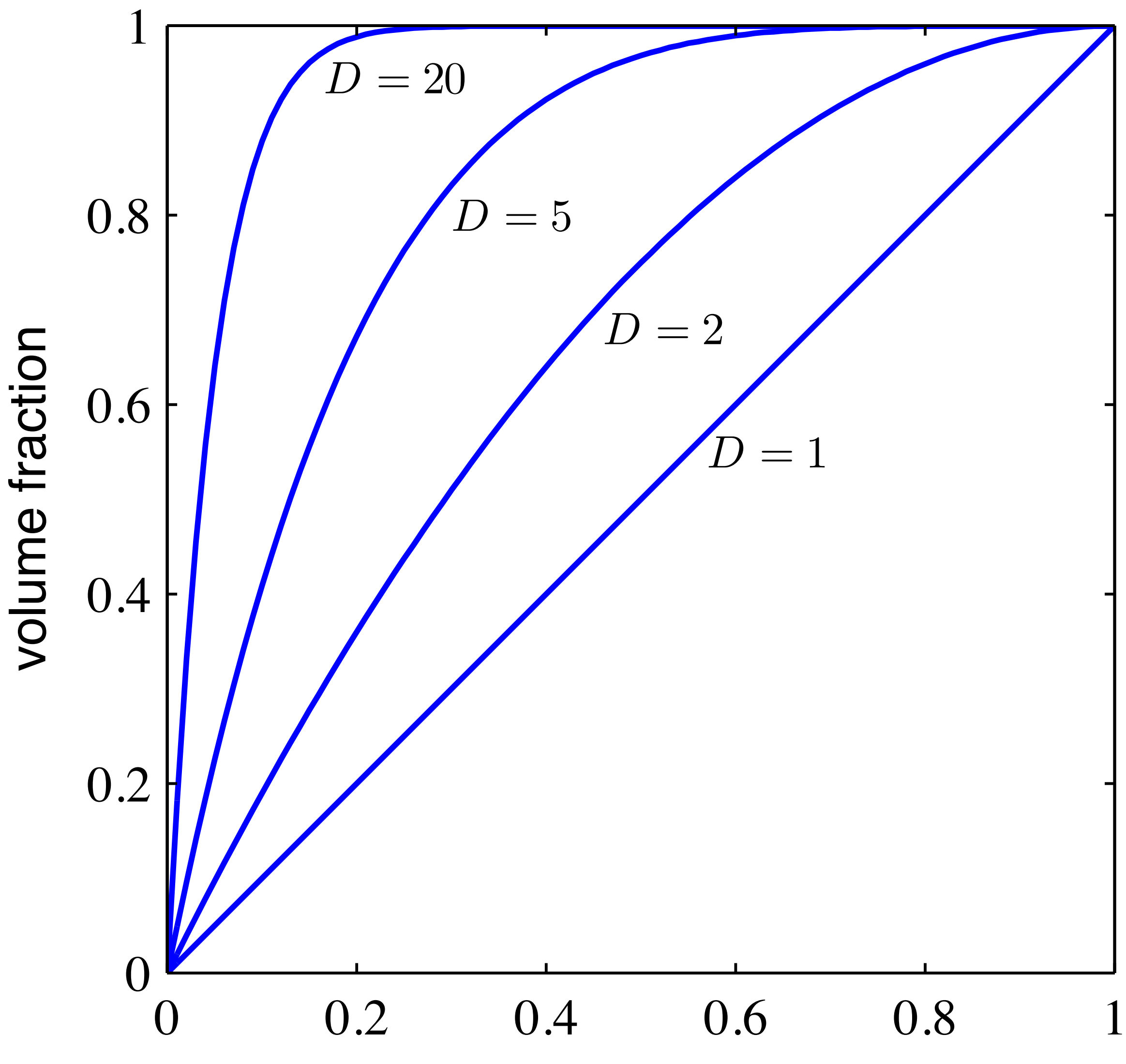
- Doesn't work well when large number of irrelevant features
  - Distances overwhelmed by noisy features

- Curse of Dimensionality
  - Distances become meaningless in high dimensions
  - (See proof in next lecture)

Curse of Dimensionality

- Consider: Sphere of radius 1 in d-dims
- Consider: an outer ε-shell in this sphere
- What is $\frac{\textit{shell volume}}{\textit{sphere volume}}$?

Curse of Dimensionality



What you need to know

- k-NN
  - Simplest learning algorithm
  - With sufficient data, very hard to beat "strawman" approach
  - Picking k?

- Kernel regression/classification
  - Set k to n (number of data points) and chose kernel width
  - Smoother than k-NN

- Problems with k-NN
  - Curse of dimensionality
  - Irrelevant features often killers for instance-based approaches
  - Slow NN search: Must remember (very large) dataset for prediction

What you need to know

- Key Concepts (which we will meet again)
  - Supervised Learning
  - Classification/Regression
  - Loss Functions
  - Statistical Estimation
  - Training vs Testing Data
  - Hypothesis Class
  - Overfitting, Generalization