BBM406
Fundamentals of Machine Learning

Lecture 3:
Kernel Regression, Distance Metrics, Curse of Dimensionality

Aykut Erdem // Hacettepe University // Fall 2019

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Administrative

- **Assignment 1** will be out Friday!

- It is due **November 1** (i.e. in two weeks).

- It includes
  - Pencil-and-paper derivations
  - Implementing kernel regression
  - numpy/Python code

Movie Recommendation System

- MovieLens dataset (100K ratings of 9K movies by 600 users)

- You may want to split the training set into train and validation (more on this next week)

- The data consists of three tables:
  - Ratings: userId, movieId, rating, timestamp
  - Movies: movieId, title, genre
  - Links: movieId, imdbId, and tmdbId
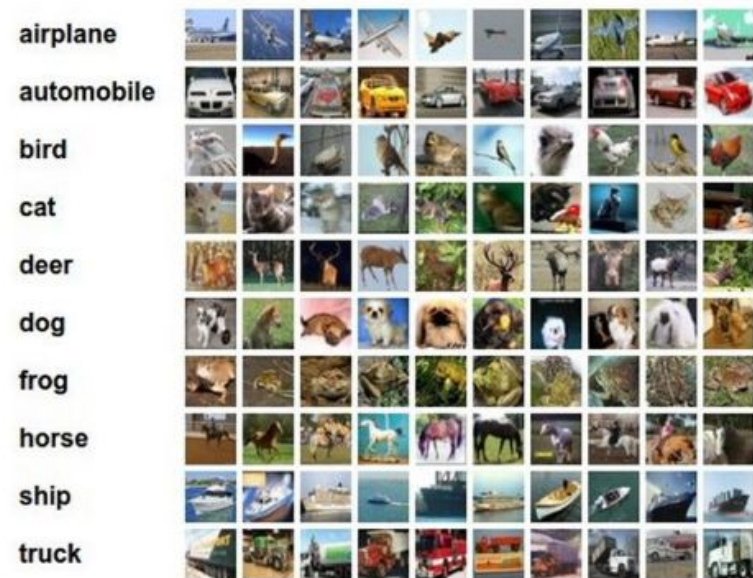 - Tags: userId, movieId, tag, timestamp

Recall from last time… **Nearest Neighbors**

Example dataset: **CIFAR-10**
10 labels
50,000 training images
10,000 test images.

airplane
automobile
bird
cat
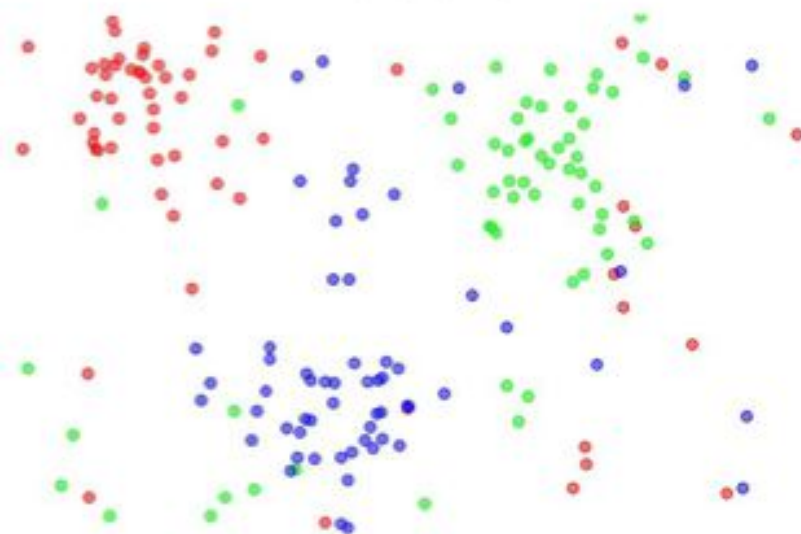deer
dog
frog
horse
ship
truck

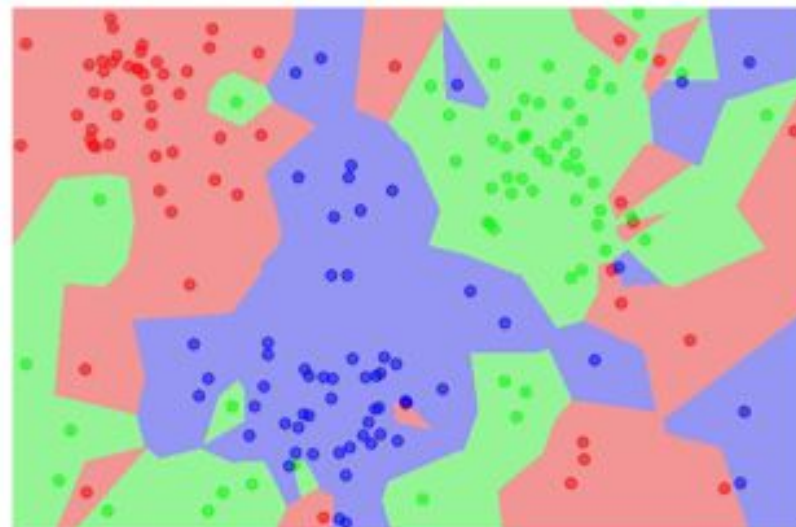For every test image (first column), examples of nearest neighbors in rows

- Very simple method

- Retain all training data
  - It can be slow in testing
  - Finding NN in high dimensions is slow

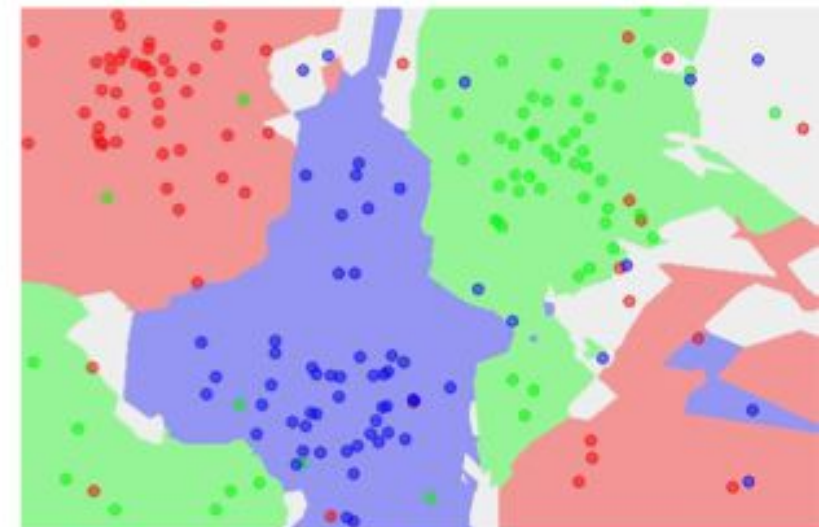- Metrics are very important

- Good baseline

the data

NN classifier

5-NN classifier

Classification

- Input: X
  - Real valued, vectors over real.
  - Discrete values (0,1,2,…)
  - Other structures (e.g., strings, graphs, etc.)
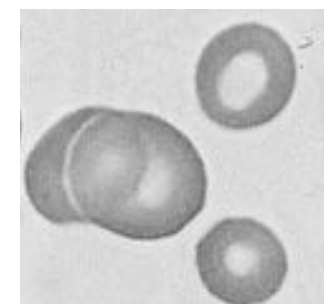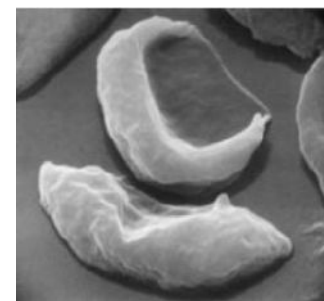
- Output: Y
  - Discrete (0,1,2,...)

X = Document → Y = Topic

Sports
Science
News

X = Cell Image → Y = Diagnosis
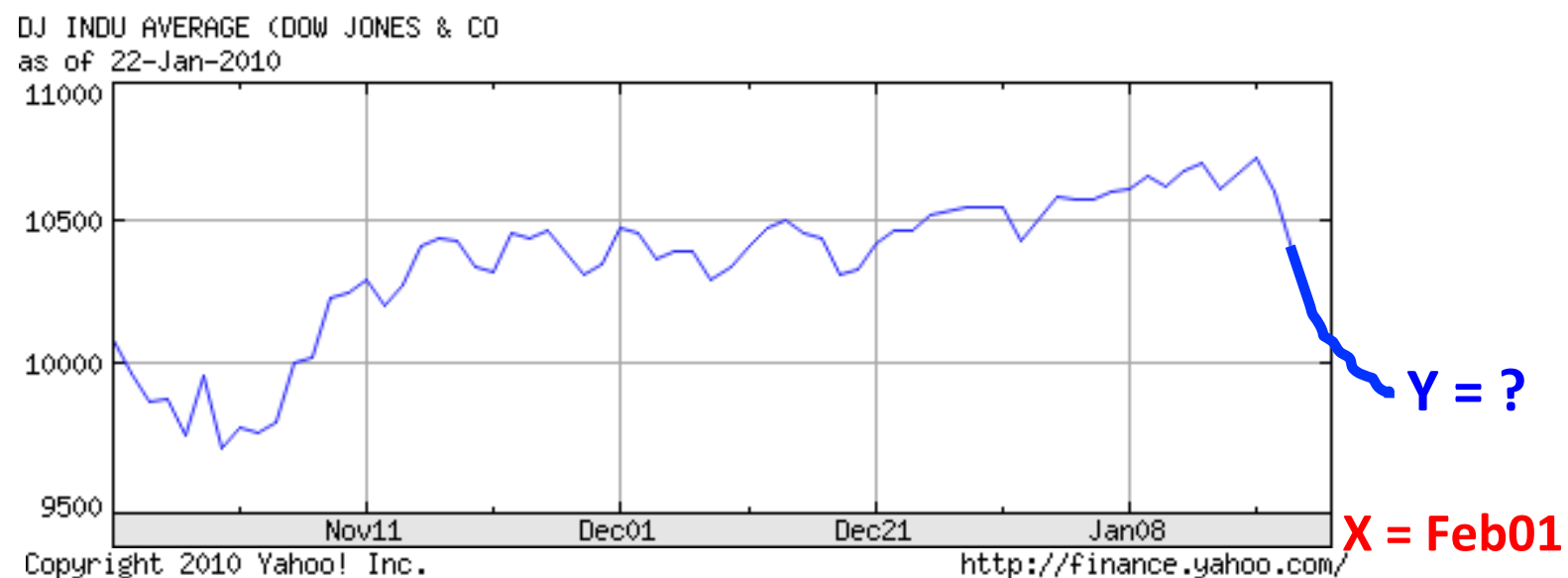
Anemic cell
Healthy cell

Regression

- Input: X

  - Real valued, vectors over real.

  - Discrete values (0,1,2,…)

  - Other structures (e.g., strings, graphs, etc.)

- Output: Y

  - Real valued, vectors over real.

Stock Market Prediction

DJ INDU AVERAGE (DOW JONES & CO
as of 22-Jan-2010

Y = ?

X = Feb01

Copyright 2010 Yahoo! Inc.   http://finance.yahoo.com/

6

What should I watch tonight?

What should I watch tonight?

What should I watch tonight?

slide by Sanja Fidler

Today

- Kernel regression
  - nonparametric

- Distance metrics

- Linear regression (*more on Friday*)
  - parametric
  - simple model

Simple 1-D Regression
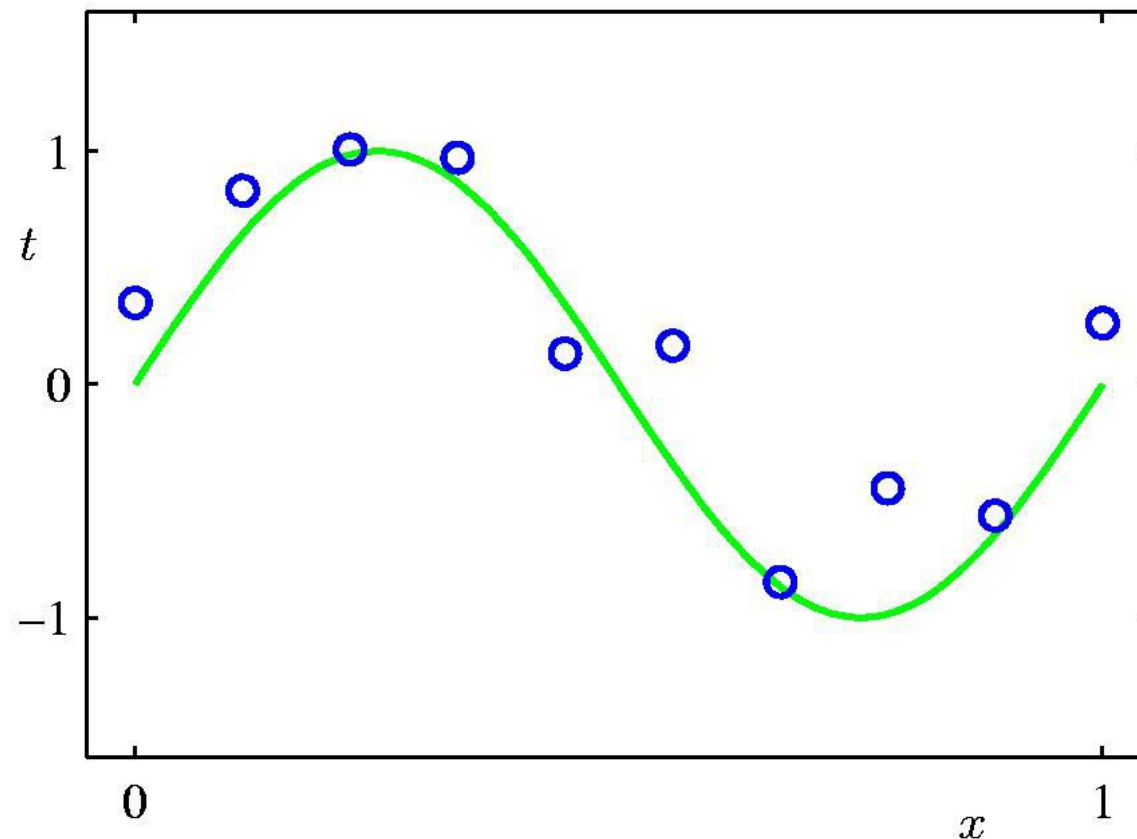
- Circles are data points (i.e., training examples) that are given to us
- The data points are uniform in x, but may be displaced in y

$$t(x) = f(x) + \varepsilon$$

with ε some noise

- In green is the "true" curve that we don't know

11

Kernel Regression

K-NN for Regression

- Given: Training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
  - Attribute vectors: $x_i \in X$
  - Target attribute $y_i \in$ $\mathcal{R}$

- Parameter:
  - Similarity function: $K : X \times X \rightarrow$ $\mathcal{R}$
  - Number of nearest neighbors to consider: k

- Prediction rule
  - New example x'
  - K-nearest neighbors: k train examples with largest $K(x_i, x')$

$$h(\vec{x}') = \frac{1}{k} \sum_{i \in knn(\vec{x}')} y_i$$

1-NN for Regression

Here, this is the closest datapoint

Here, this is the closest datapoint

Here, this is the closest datapoint

Here, this is the closest datapoint

y

x

Figure Credit: Carlos Guestrin

14

1-NN for Regression

- Often bumpy (overfits)

slide by Dhruv Batra
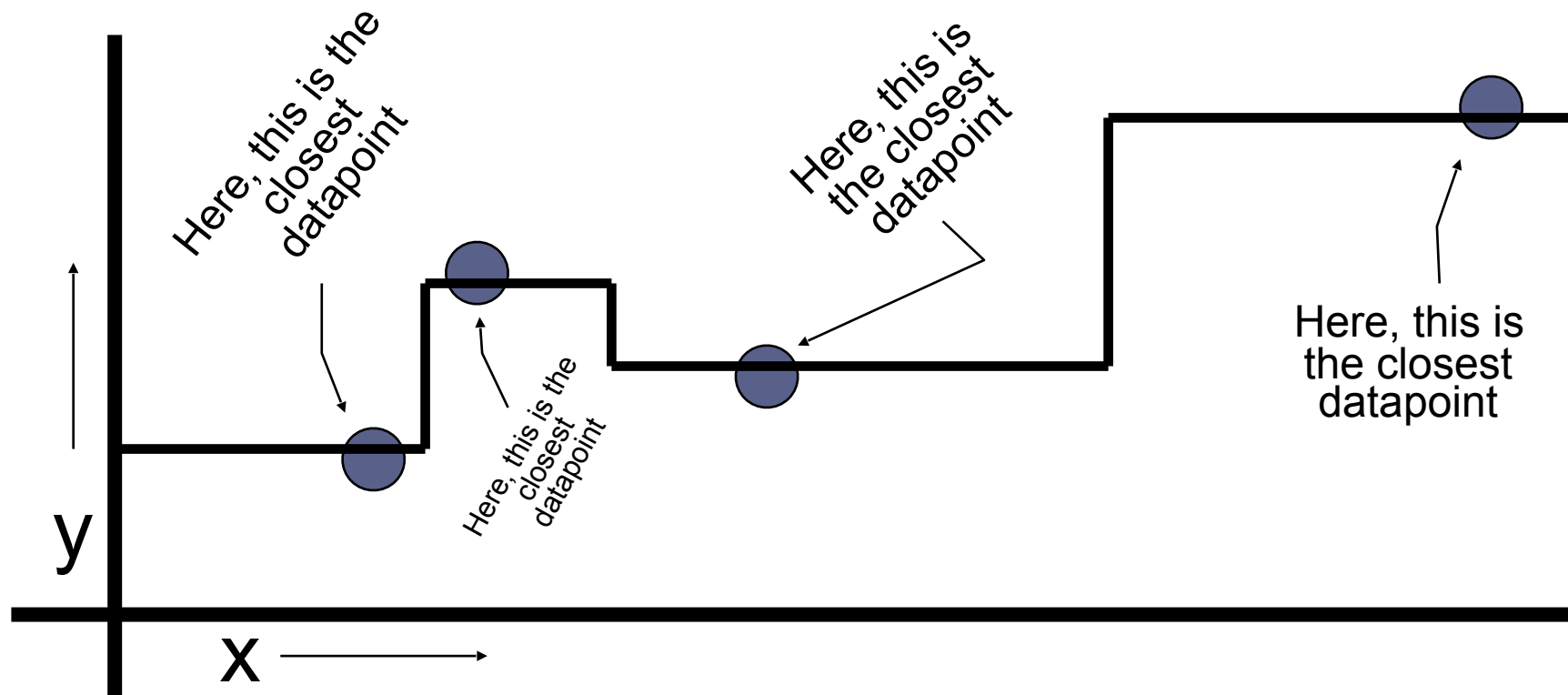
9-NN for Regression

- Often bumpy (overfits)
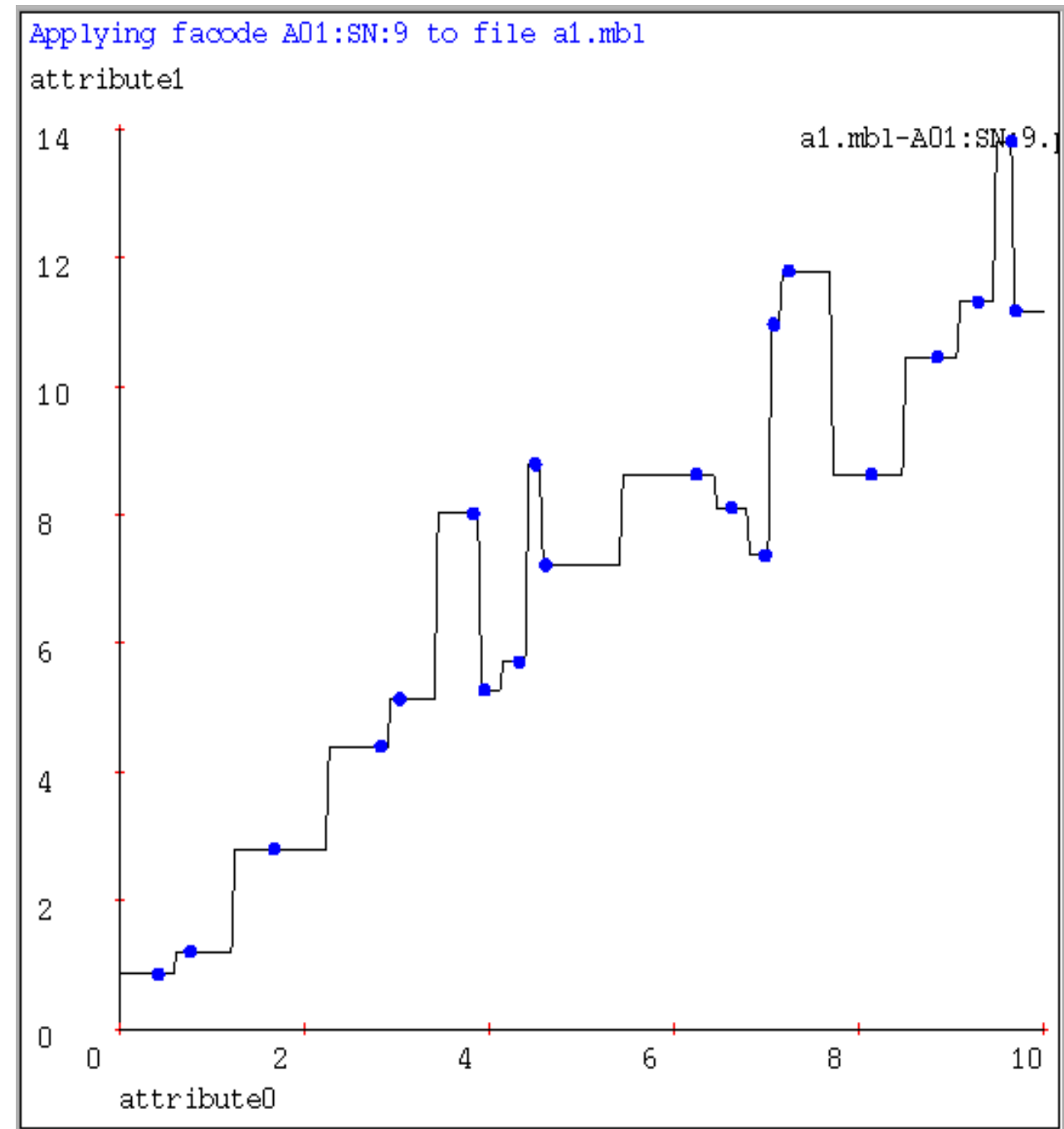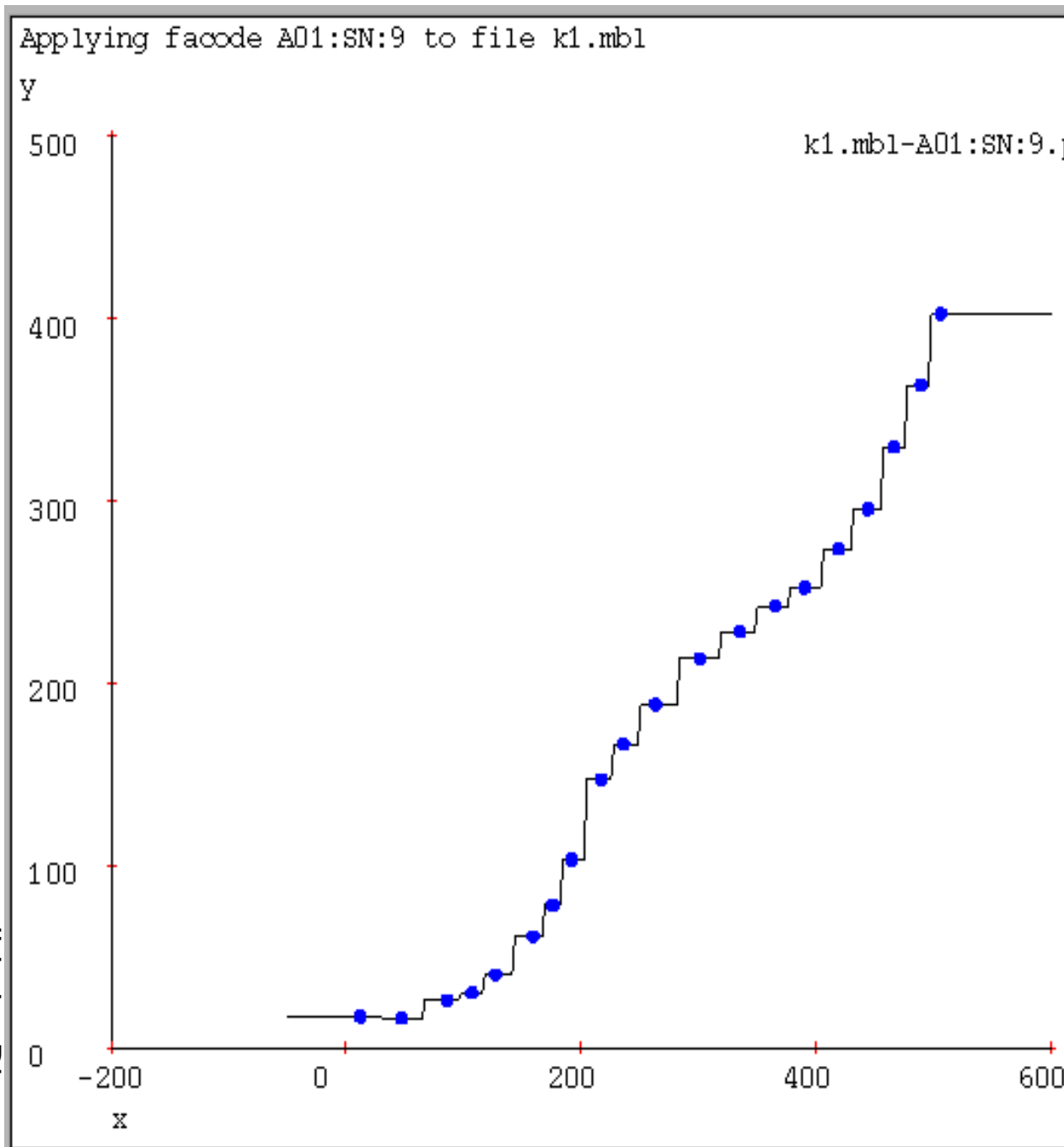
Figure Credit: Andrew Moore

16

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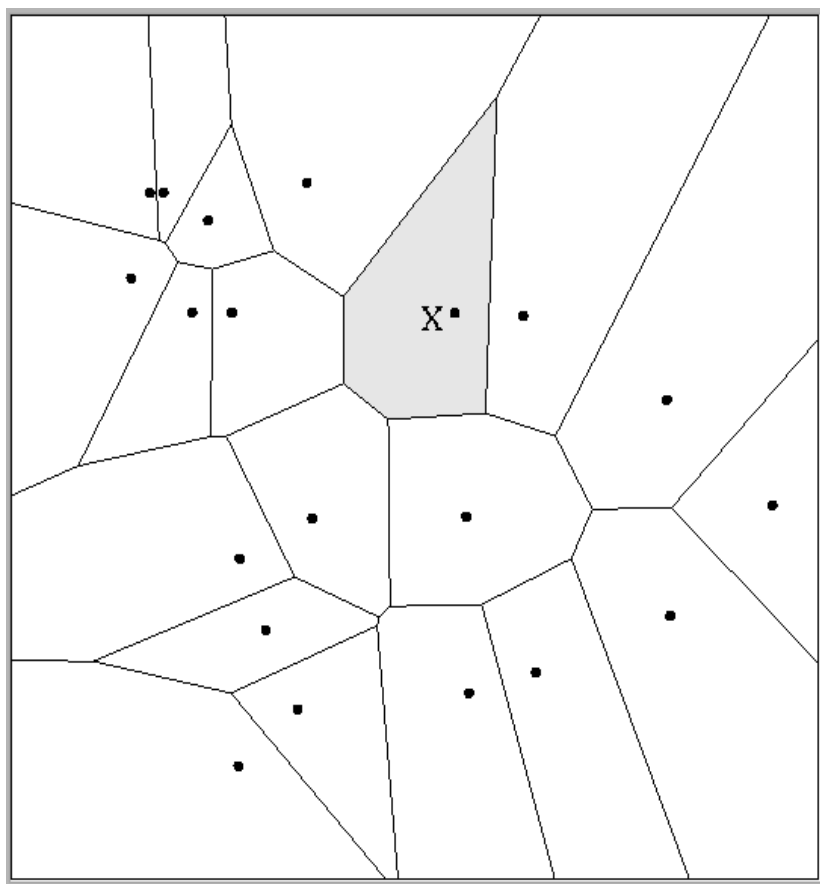
$$\text{Dist}(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2$$
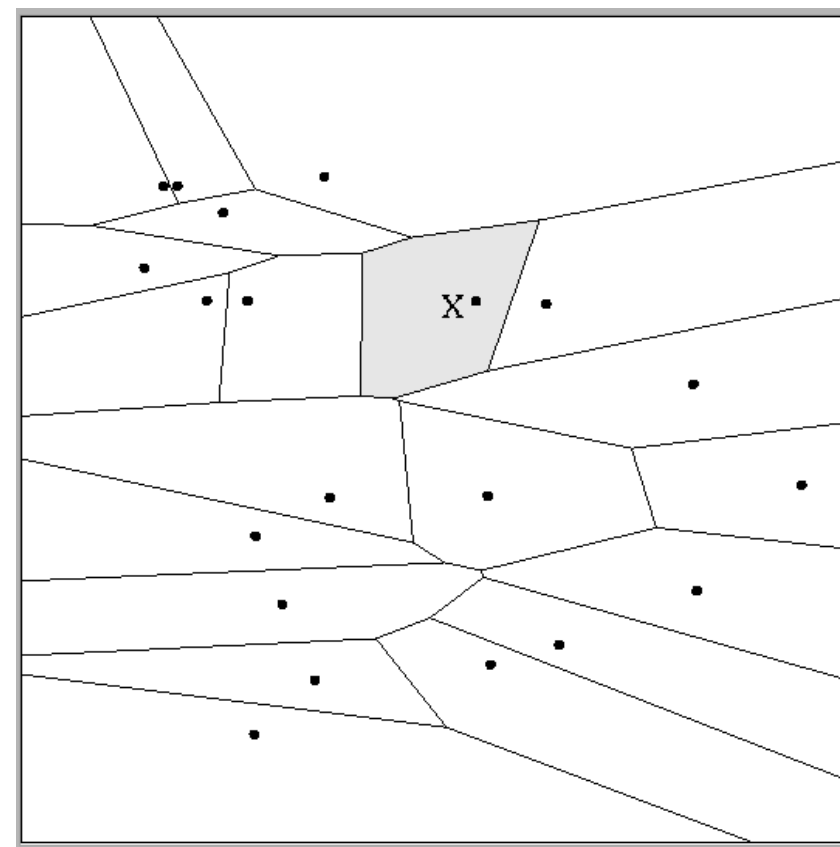
$$\text{Dist}(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (3x_{i2} - 3x_{j2})^2$$

The relative scalings in the distance metric affect region shapes

Example: Choosing a restaurant

- In everyday life we need to make decisions by taking into account lots of factors

- The question is what weight we put on each of these factors (how important are they with respect to the others).

Reviews (out of 5 stars)	$	Distance	Cuisine (out of 10)
4	30	21	7
2	15	12	8
5	27	53	9
3	20	5	6

?

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)
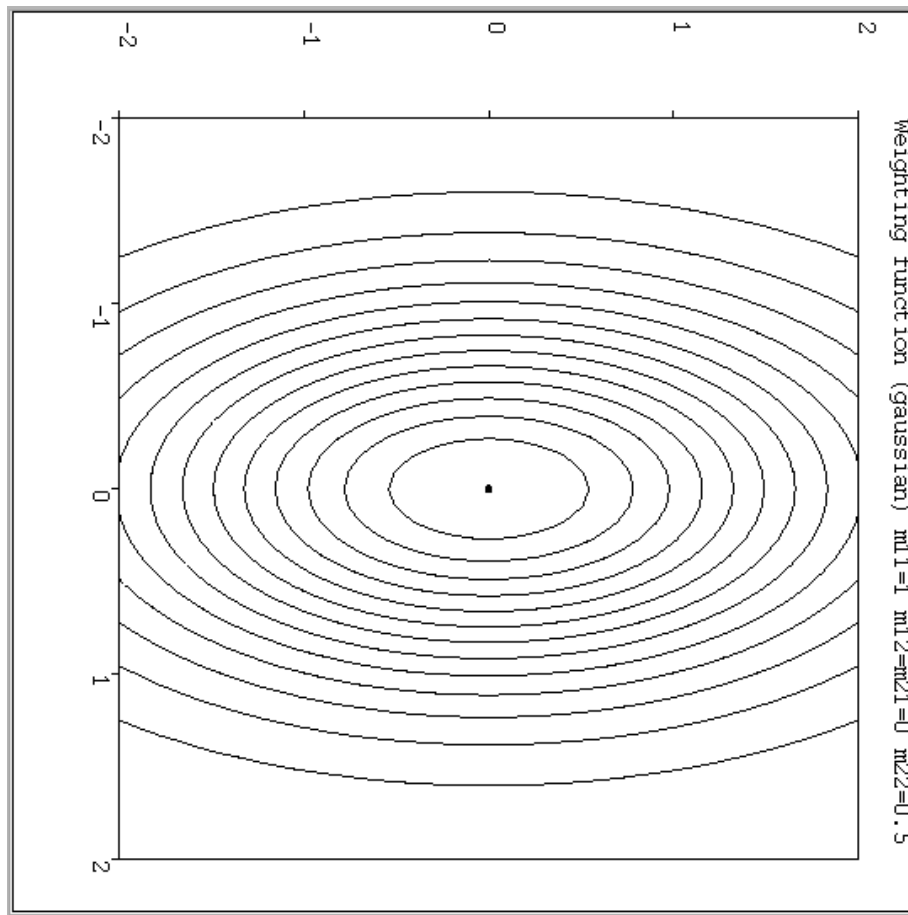
Scaled Euclidian (L$_2$)
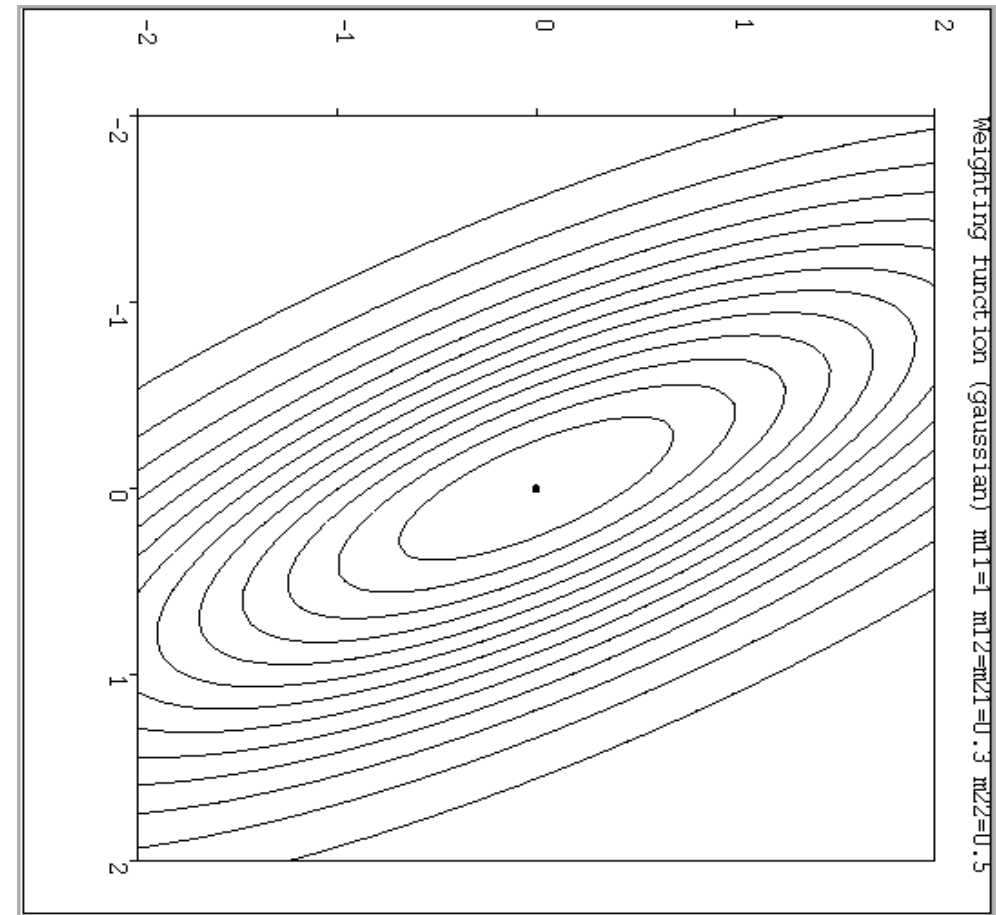
Mahalanobis (non-diagonal A)

Slide Credit: Carlos Guestrin

Minkowski distance

$p = 2^{-2}$
$= 0.25$

$p = 2^{-1.5}$
$= 0.354$

$p = 2^{-1}$
$= 0.5$

$p = 2^{-0.5}$
$= 0.707$

$p = 2^{0}$
$= 1$

$p = 2^{0.5}$
$= 1.414$

$p = 2^{1}$
$= 2$

$p = 2^{1.5}$
$= 2.828$

$p = 2^{2}$
$= 4$

$\ldots$

$p = 2^{\infty}$
$= \infty$
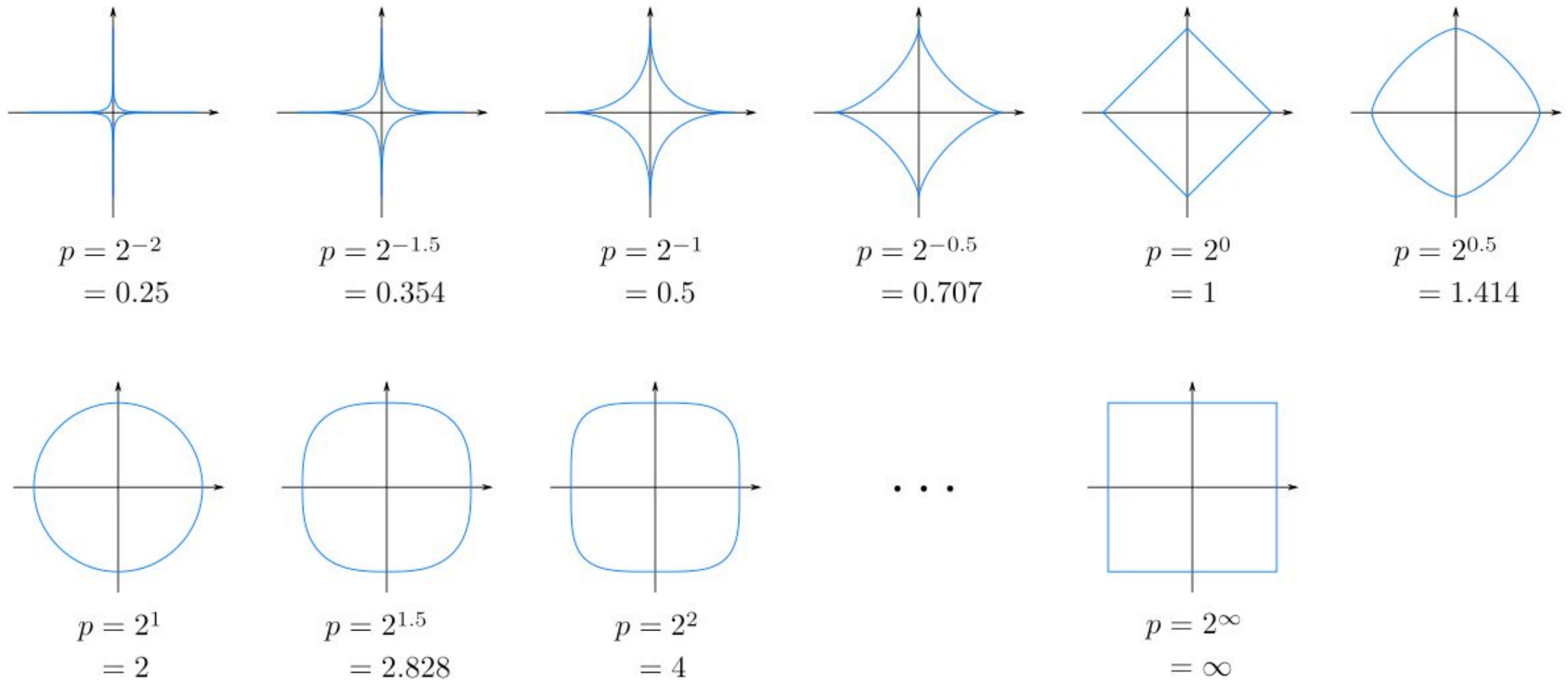
$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Notable distance metrics (and their level sets)
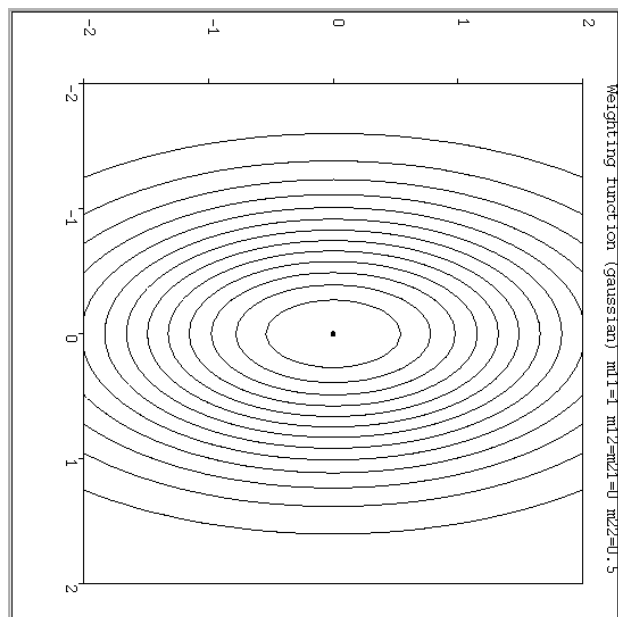
Scaled Euclidian (L_2)
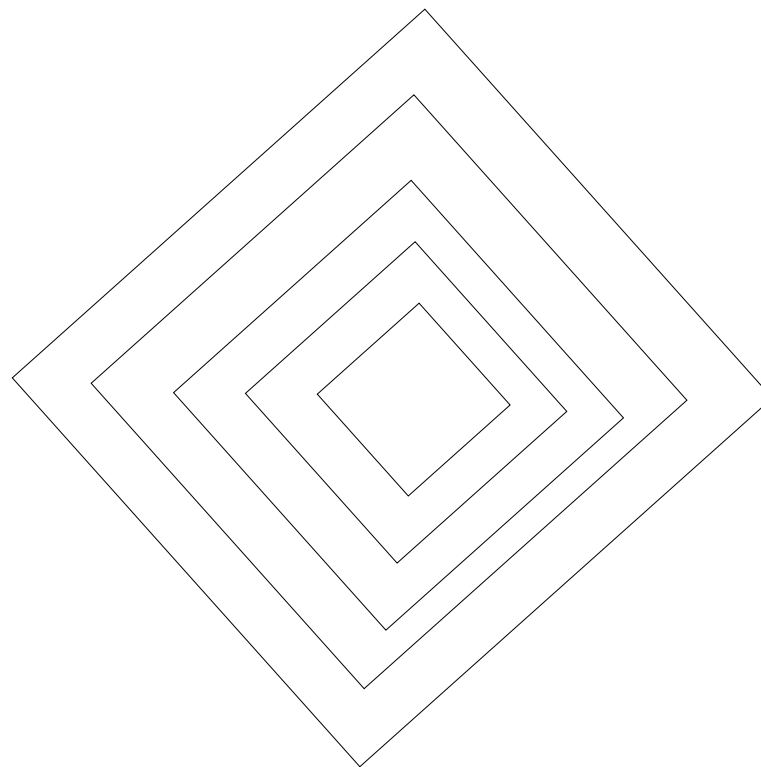
L_1 norm (absolute)

L_{inf} *(max) norm*

Slide Credit: Carlos Guestrin
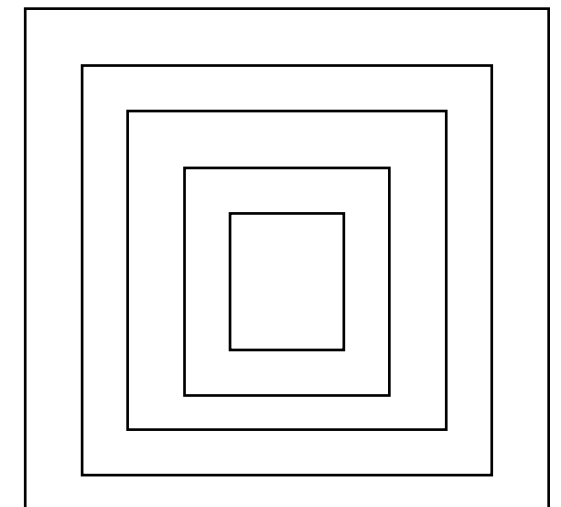
Weighted K-NN for Regression

- Given: Training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
  - Attribute vectors: $x_i \in X$
  - Target attribute $\quad y_i \in \qquad \mathcal{R}$

- Parameter:
  - Similarity function: $K : X \times X \rightarrow \qquad \mathcal{R}$
  - Number of nearest neighbors to consider: k

- Prediction rule
  - New example x'
  - K-nearest neighbors: k train examples with largest $K(x_i, x')$

$$h(\vec{x}') = \frac{\sum_{i \in knn(\vec{x}')} y_i K(\vec{x}_i, \vec{x}')}{\sum_{i \in knn(\vec{x}')} K(\vec{x}_i, \vec{x}')}$$

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 = $\Sigma w_i y_i / \Sigma w_i$

Slide Credit: Carlos Guestrin

Weighting/Kernel functions

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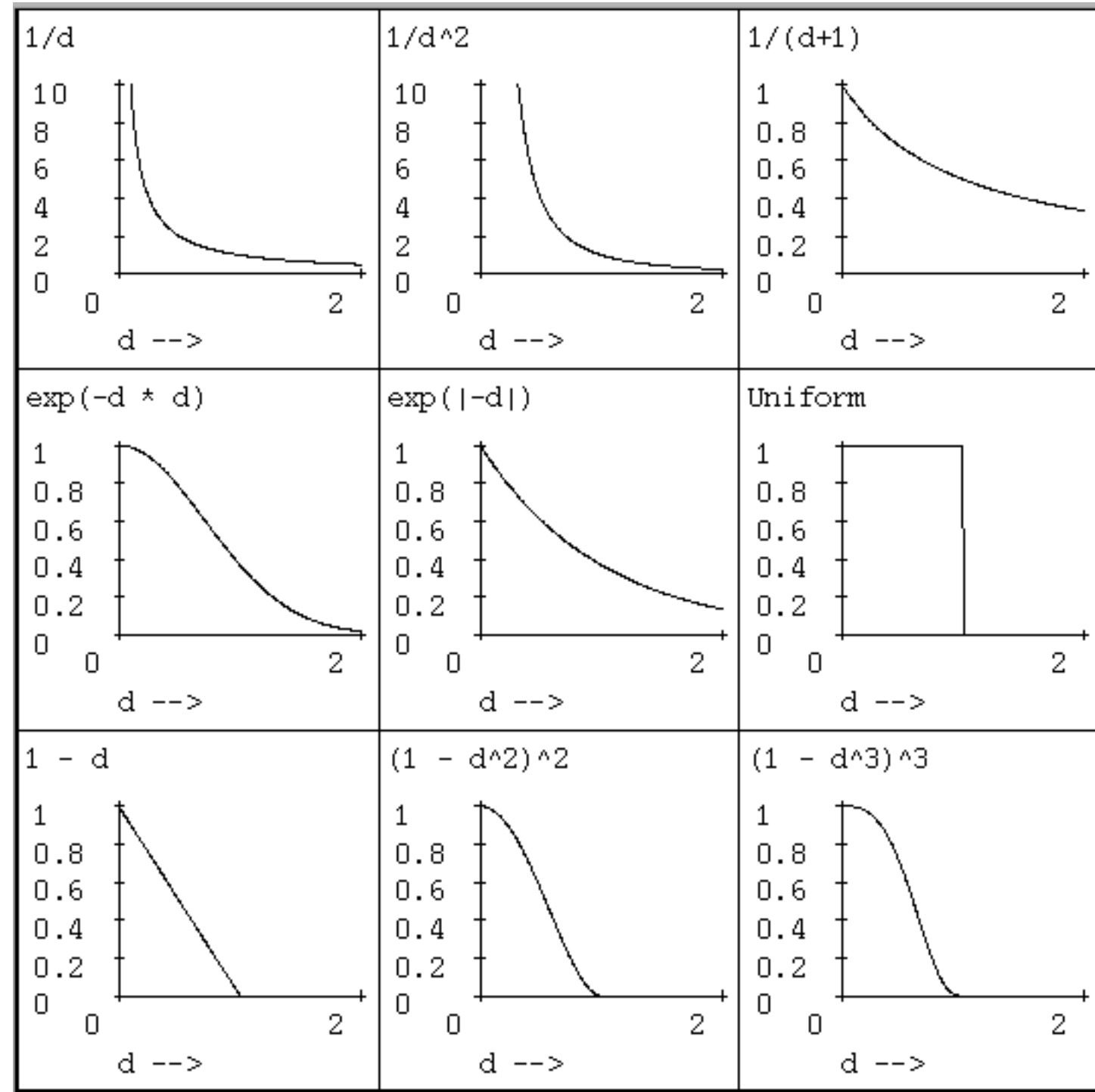
(Our examples use Gaussian)

Slide Credit: Carlos Guestrin

Effect of Kernel Width

- What happens as $\sigma \to inf$?

- What happens as $\sigma \to 0$?
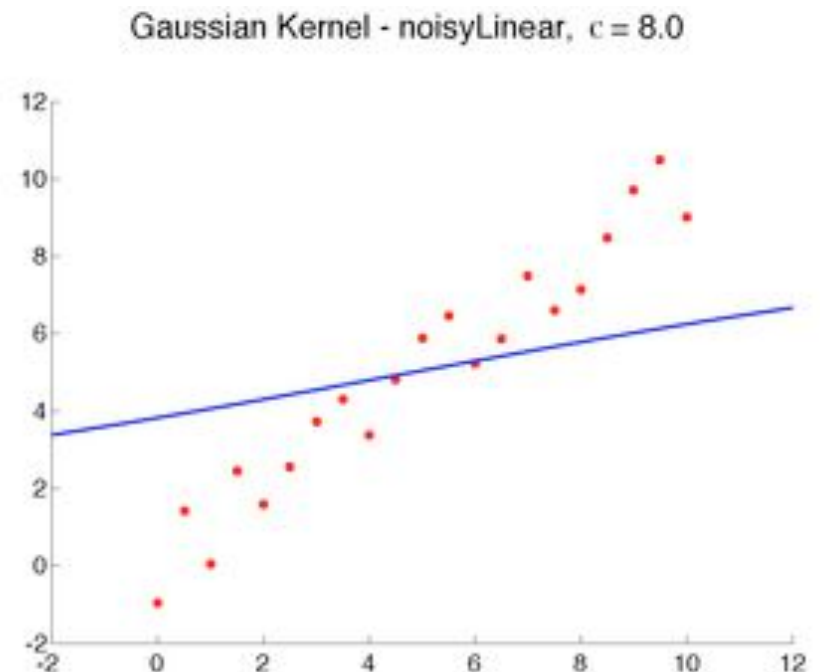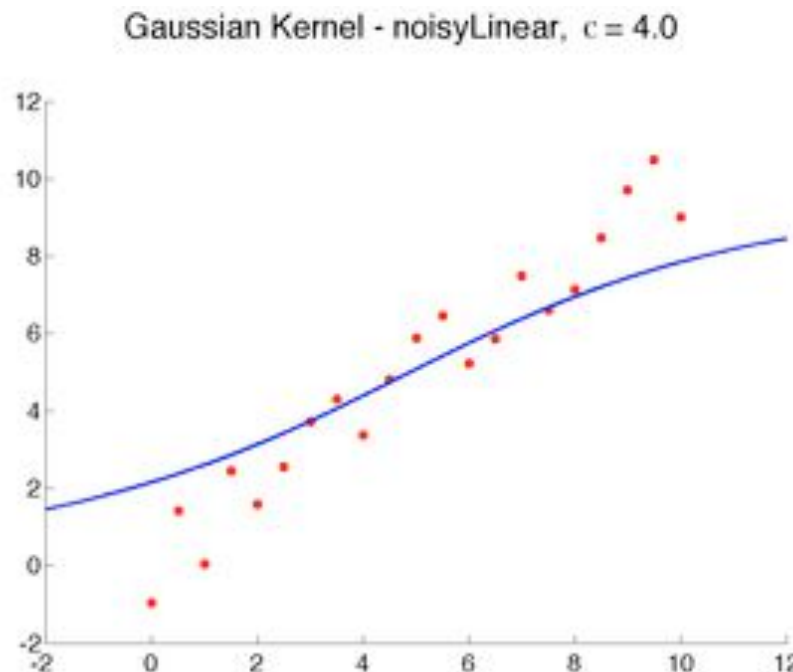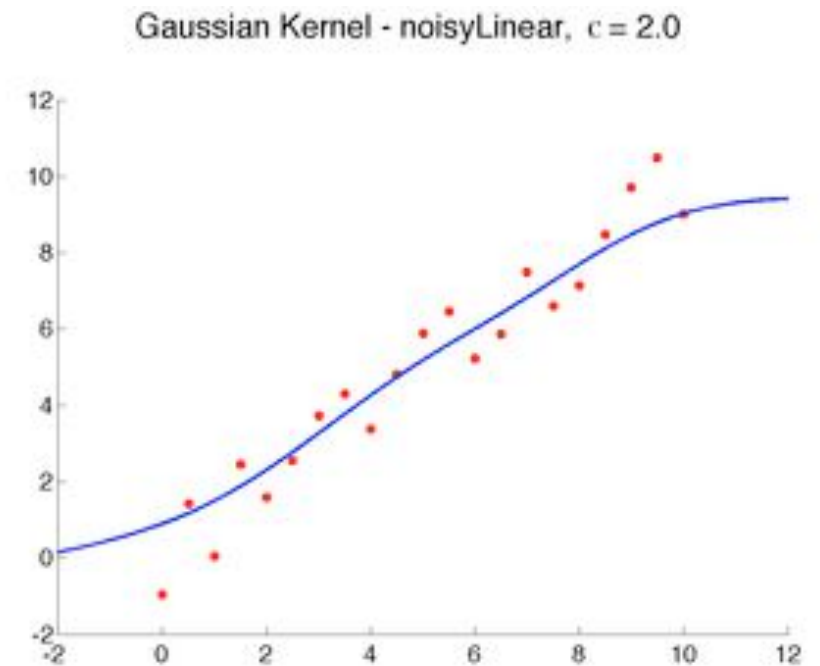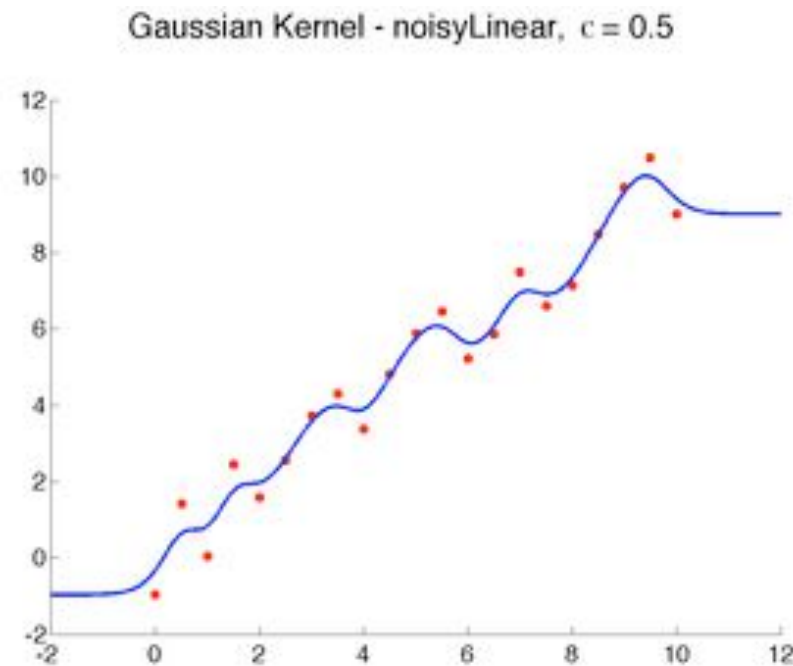
Gaussian Kernel - noisyLinear, $c = 0.5$

Gaussian Kernel - noisyLinear, $c = 2.0$

Gaussian Kernel - noisyLinear, $c = 4.0$

Gaussian Kernel - noisyLinear, $c = 8.0$

slide by Dhruv Batra

Image Credit: Ben Taskar

26

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

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

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

Curse of Dimensionality

- Consider applying a KNN classifier/regressor to data where the inputs are uniformly distributed in the D-dimensional unit cube.

- Suppose we estimate the density of class labels around a test point x by "growing" a hyper-cube around x until it contains a desired fraction f of the data points.

- The expected edge length of this cube will be $e_D(f) = f^{1/D}$.

- If $D = 10$, and we want to base our estimate on 10% of the data, we have $e_{10}(0.1) = 0.8$, so we need to extend the cube 80% along each dimension around x.

- Even if we only use 1% of the data, we find $e_{10}(0.01) = 0.63$. — no longer very local

30
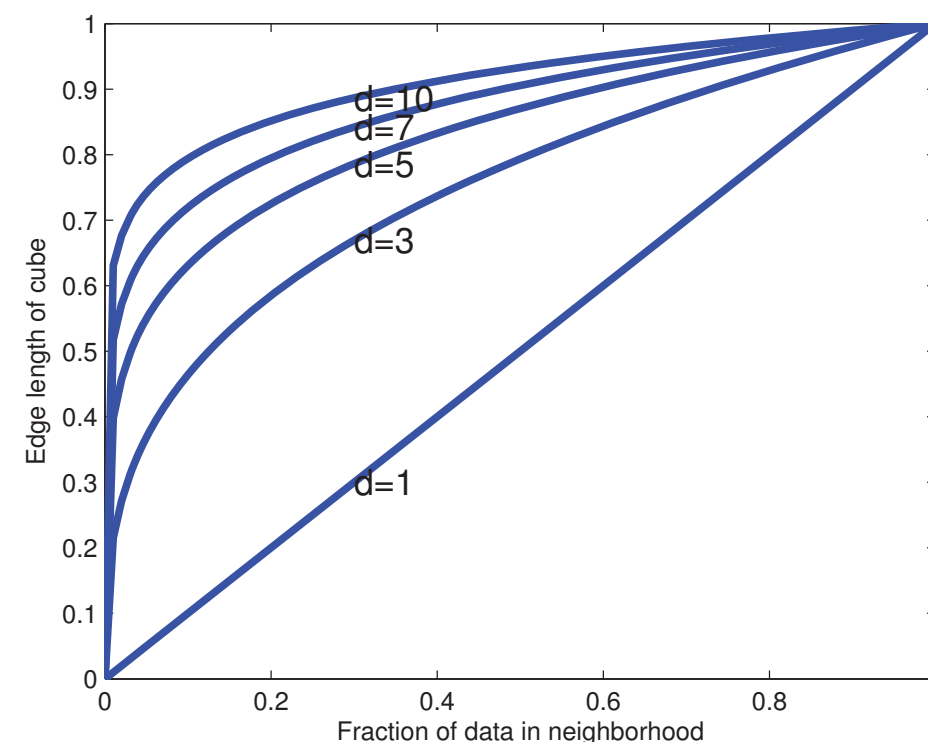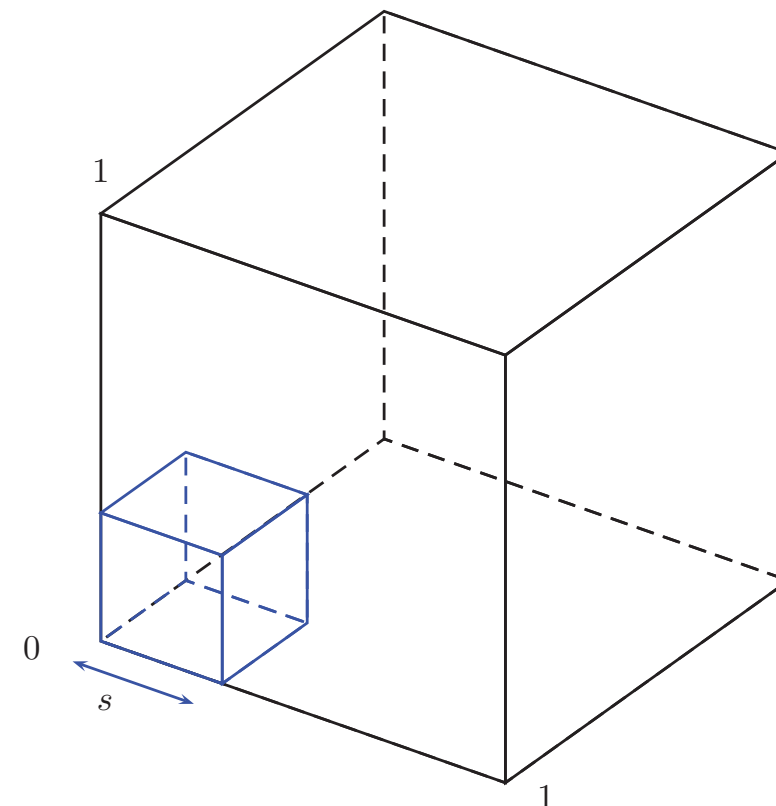
Parametric vs Non-parametric Models

- Does the capacity (size of hypothesis class) grow with size of training data?
  - Yes = Non-parametric Models
  - No = Parametric Models

Next Lecture:
Linear Regression,
Least Squares Optimization,
Model complexity, Regularization