photo:@rewardyfahmi // Unsplash

AIN311

Fundamentals of Machine Learning

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

HACETTEPE UNIVERSITY COMPUTER VISION LAB

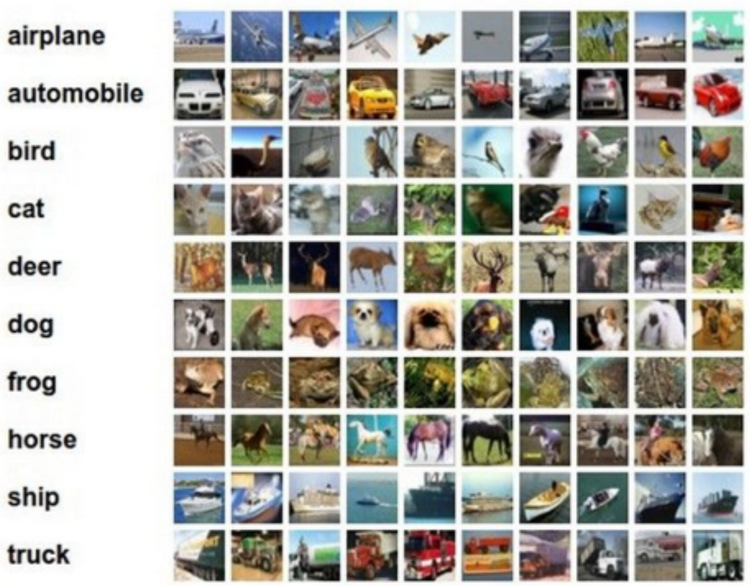
Erkut Erdem // Hacettepe University // Fall 2023

Administrative

- **Assignment 1** will be out this week!
- You will have two weeks to submit your solutions.
- It includes
  - Pencil-and-paper derivations
  - Implementing kernel regression
  - numpy/Python code

Recall from last time… **Nearest Neighbors**

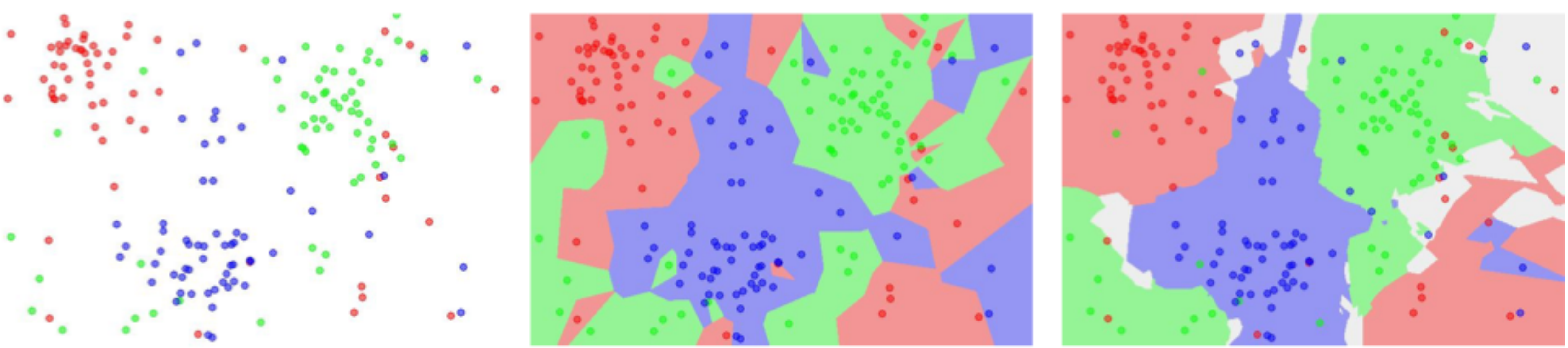
Example dataset: **CIFAR-10**
10 labels
50,000 training images
10,000 test images.



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



adopted from Fei-Fei Li & Andrej Karpathy & Justin Johnson

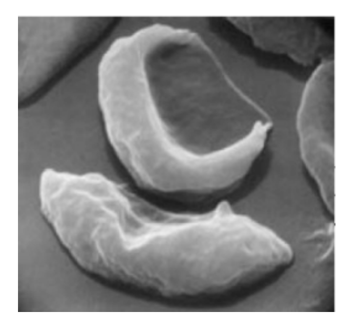
Classification

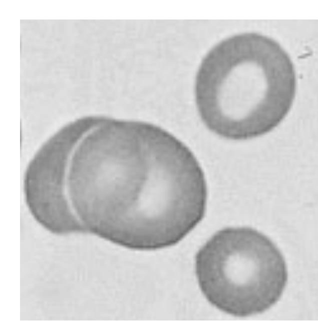
- Input: X
  - Real valued, vectors over real.
  - Discrete values (0,1,2,…)
  - Other structures (e.g., strings, graphs, etc.)
- Output: Y
  - Discrete (0,1,2,...)

Sports
Science
News

X = Document

Y = Topic





Anemic cell
Healthy cell

X = Cell Image

Y = Diagnosis

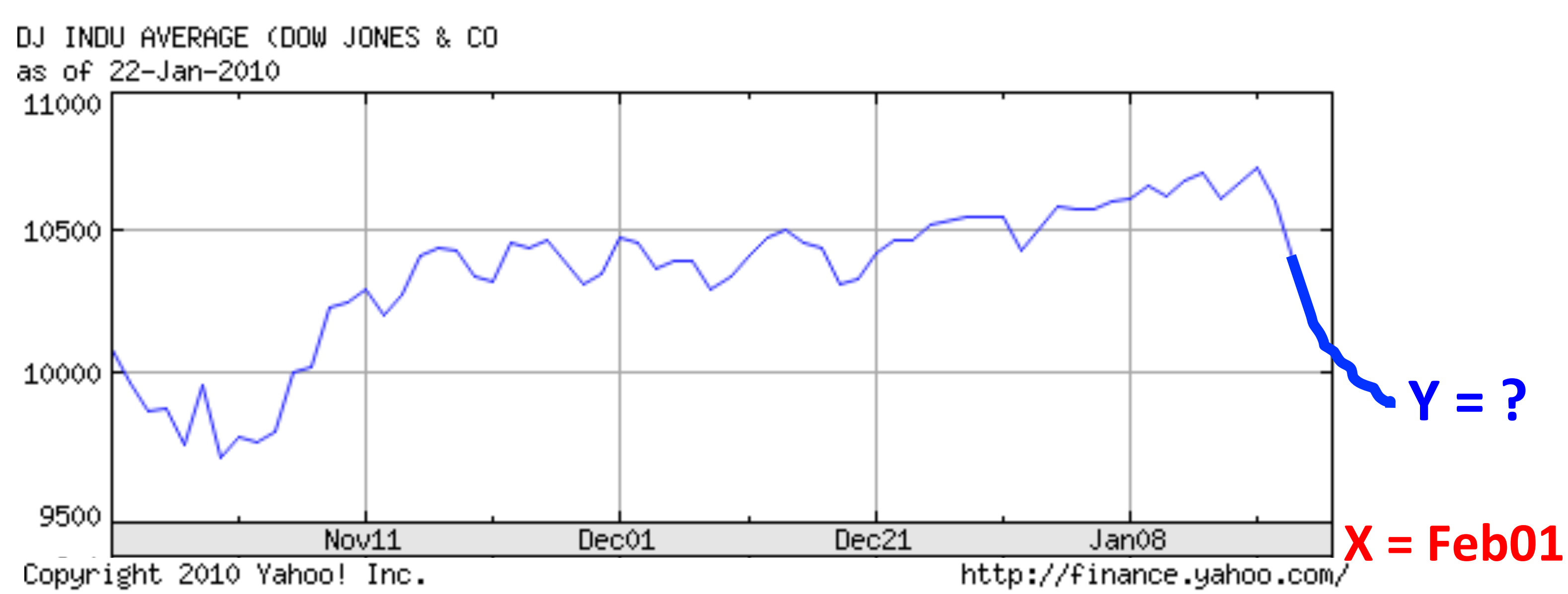
slide by Aarti Singh and Barnabas Poczos

Regression

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

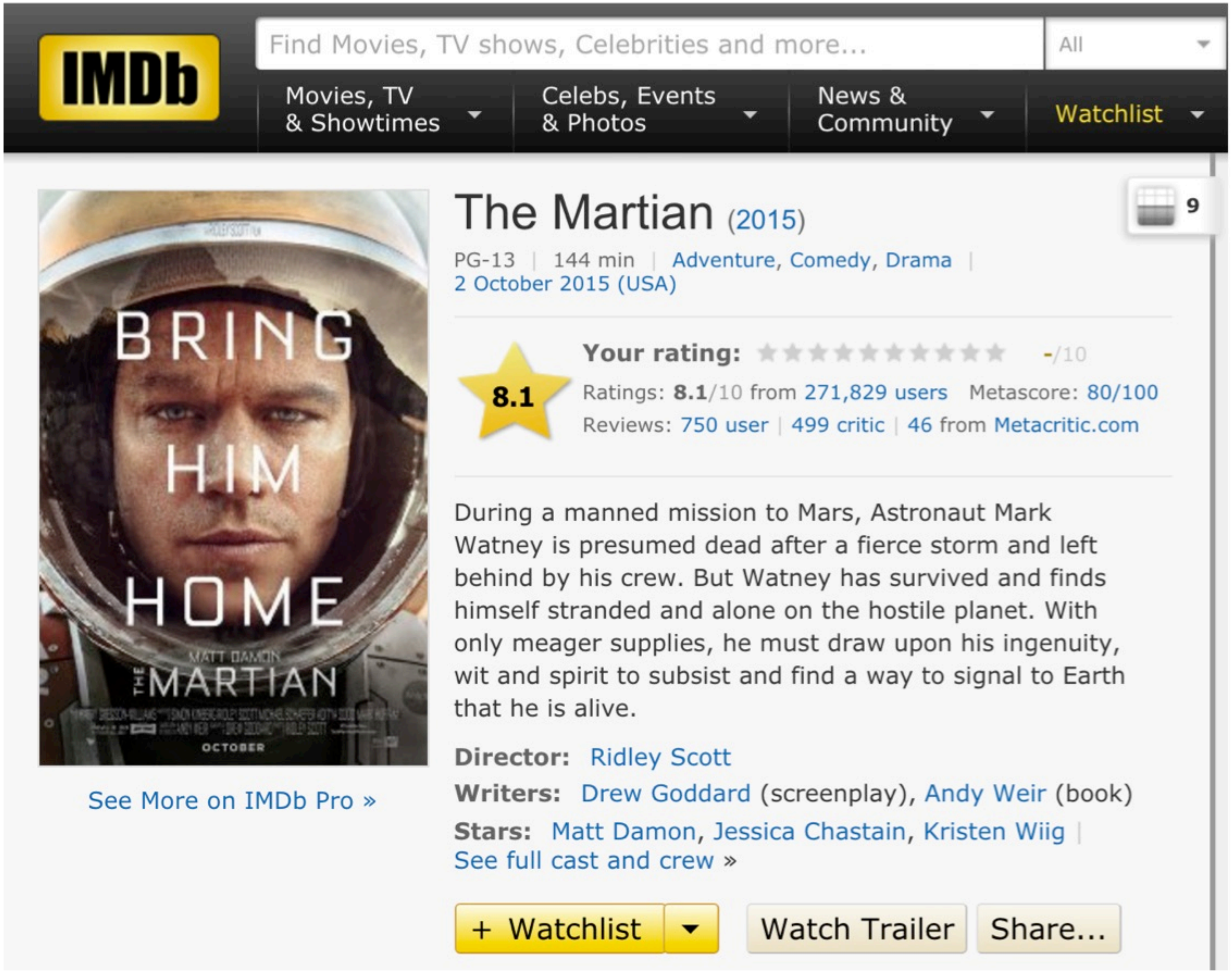
- Output: Y
  - Real valued, vectors over real.

Stock Market Prediction



slide by Aarti Singh and Barnabas Poczos

What should I watch tonight?



slide by Sanja Fidler

What should I watch tonight?

slide by Sanja Fidler

What should I watch tonight?

slide by Sanja Fidler

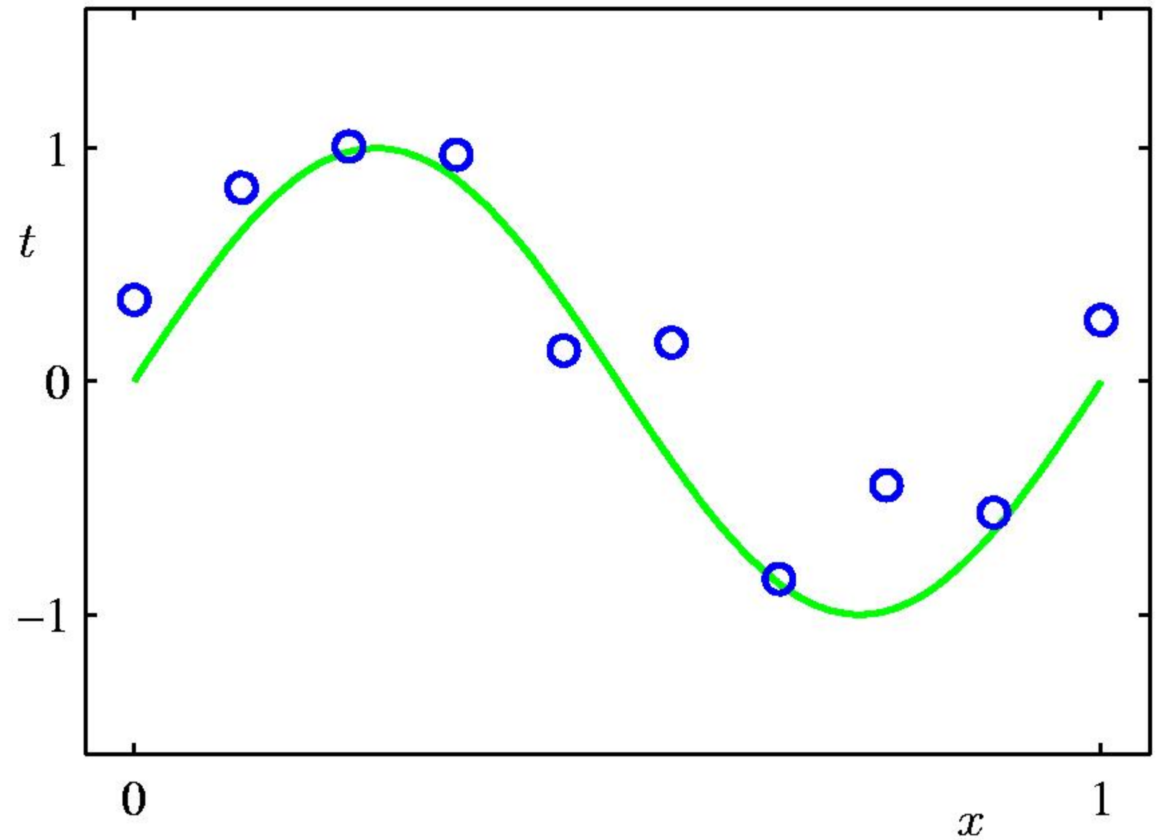
Today

- Kernel regression
  - nonparametric

- Distance metrics

- Linear regression (*more on our next lecture*)
  - 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

slide by Sanja Fidler

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 $K(x_i,x')$

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

slide by Thorsten Joachims

1-NN for Regression

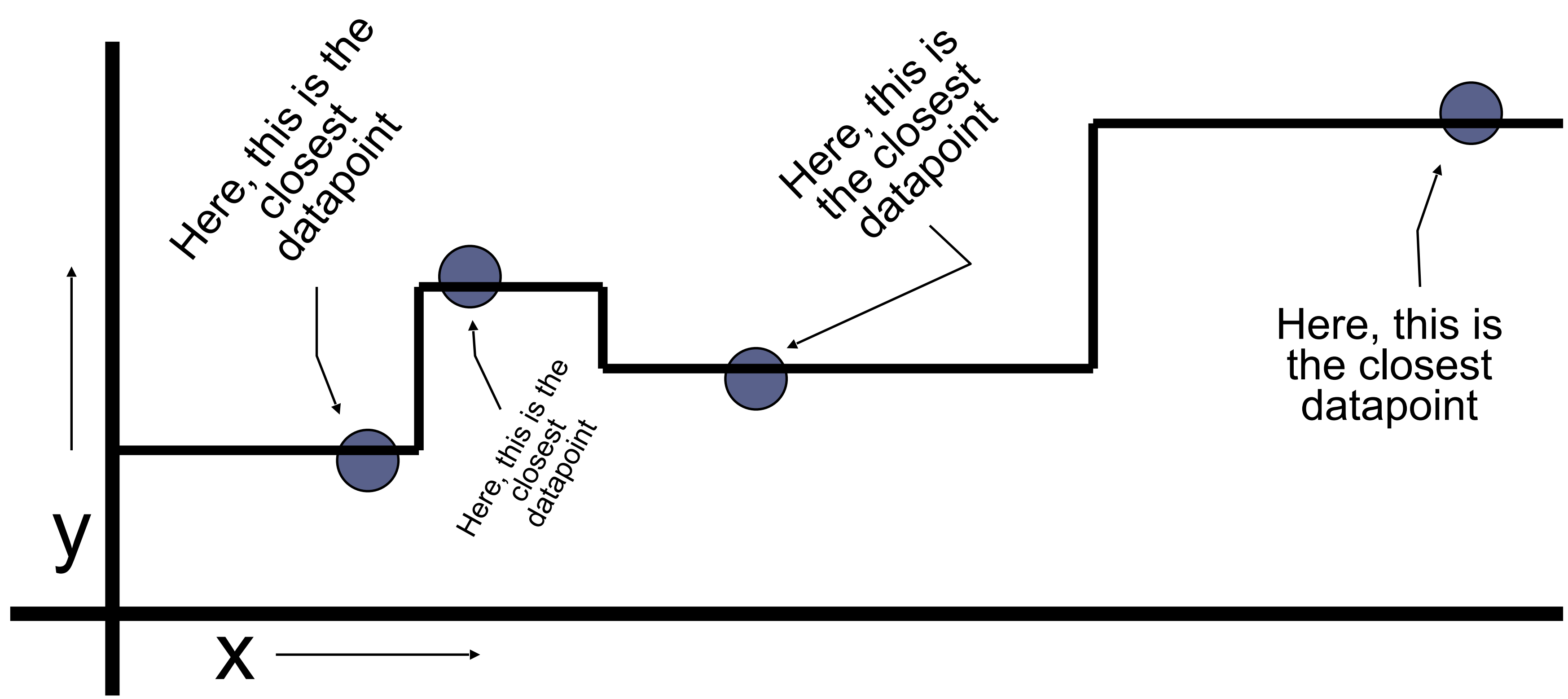
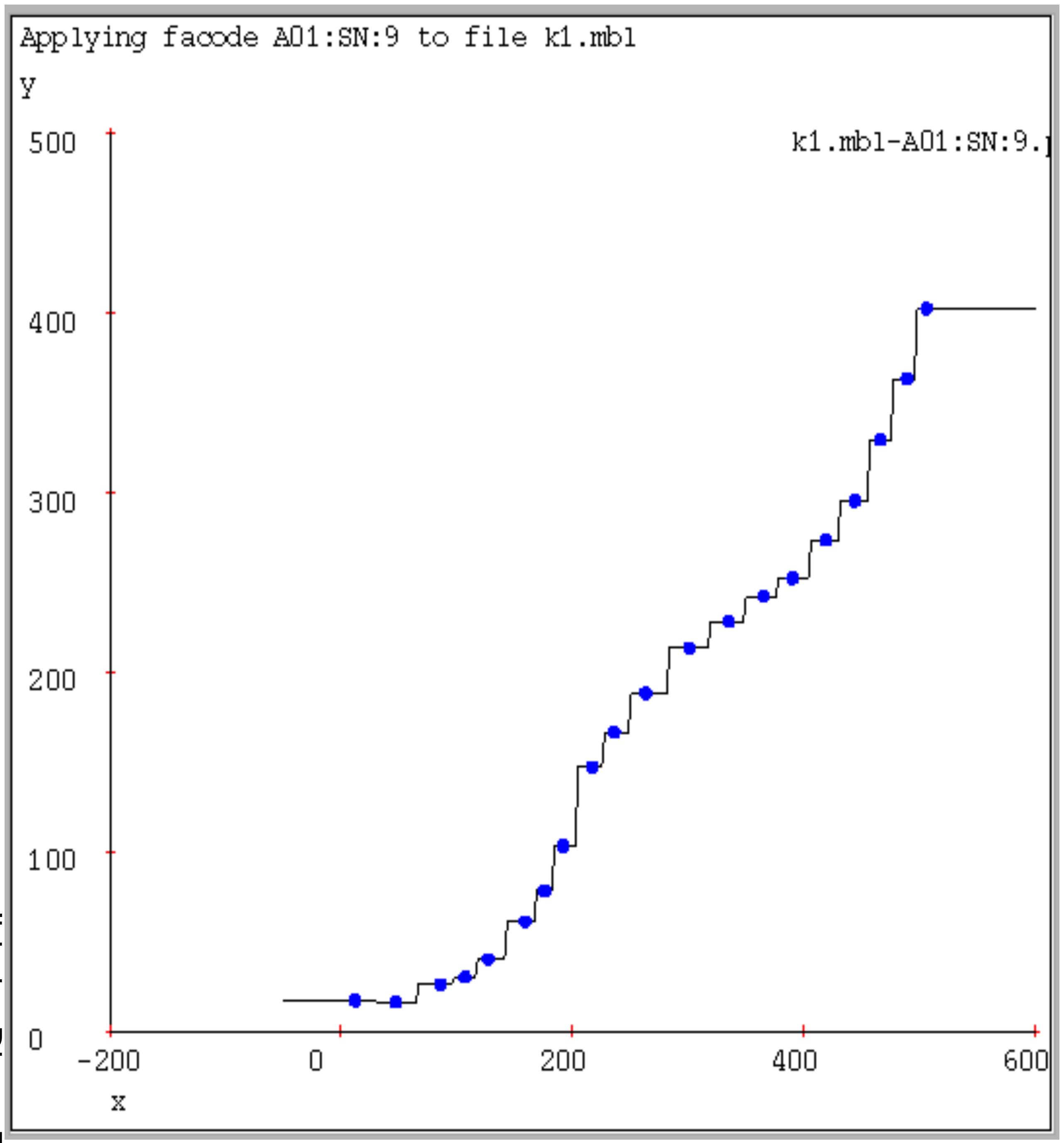


Figure Credit: Carlos Guestrin

slide by Dhruv Batra

1-NN for Regression

- Often bumpy (overfits)



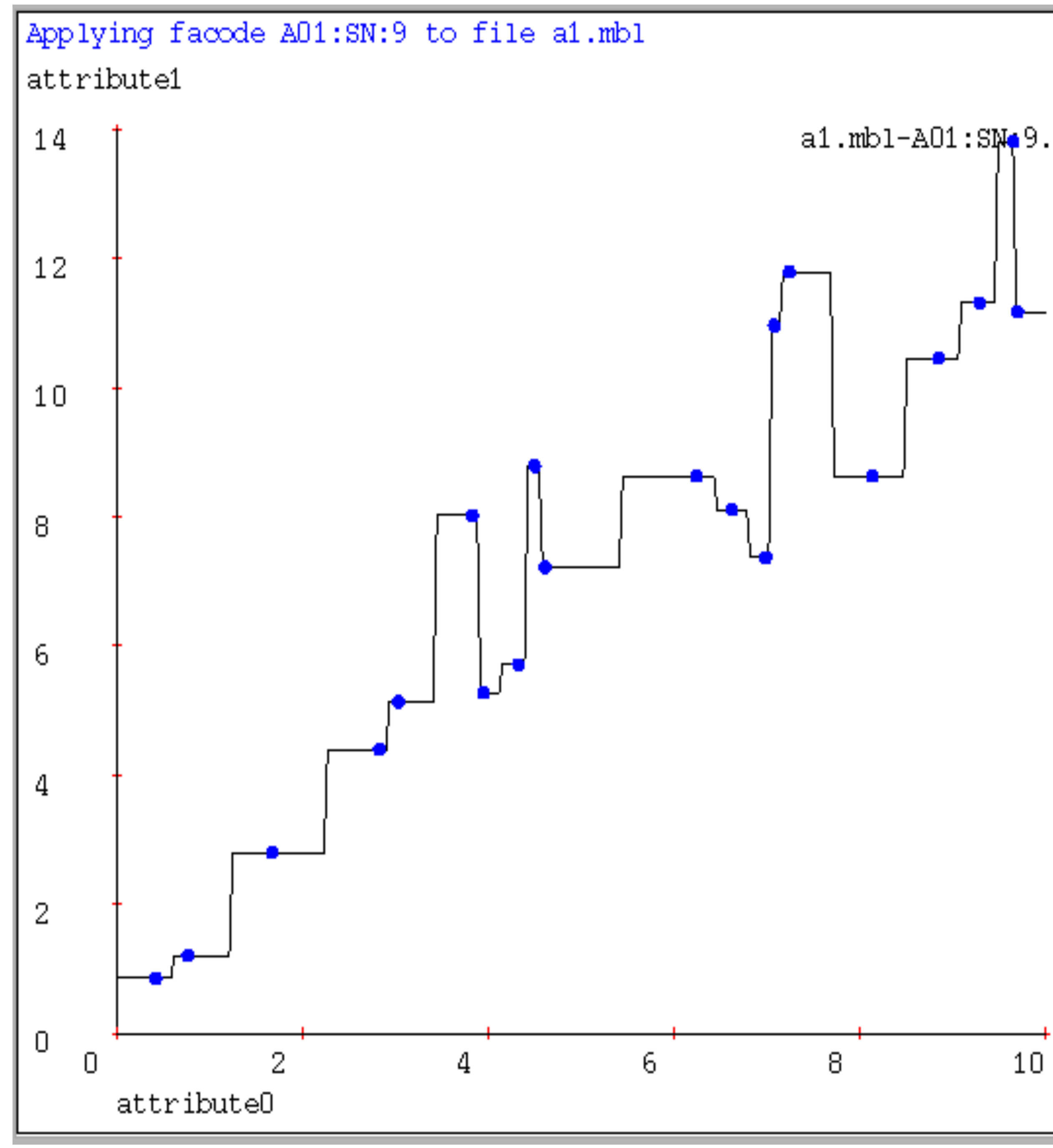
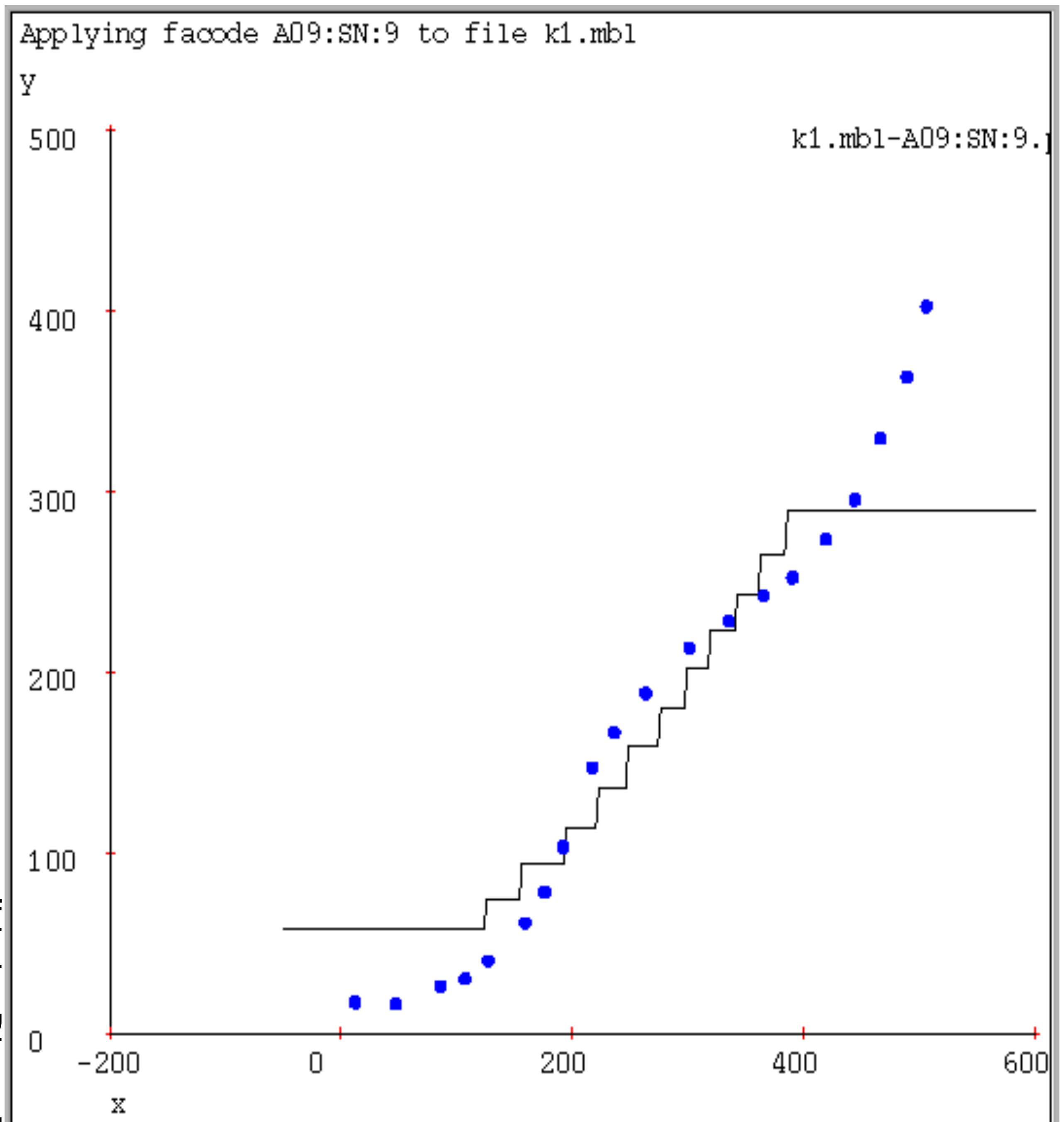


Figure Credit: Andrew Moore

slide by Dhruv Batra

9-NN for Regression

- Often bumpy (overfits)



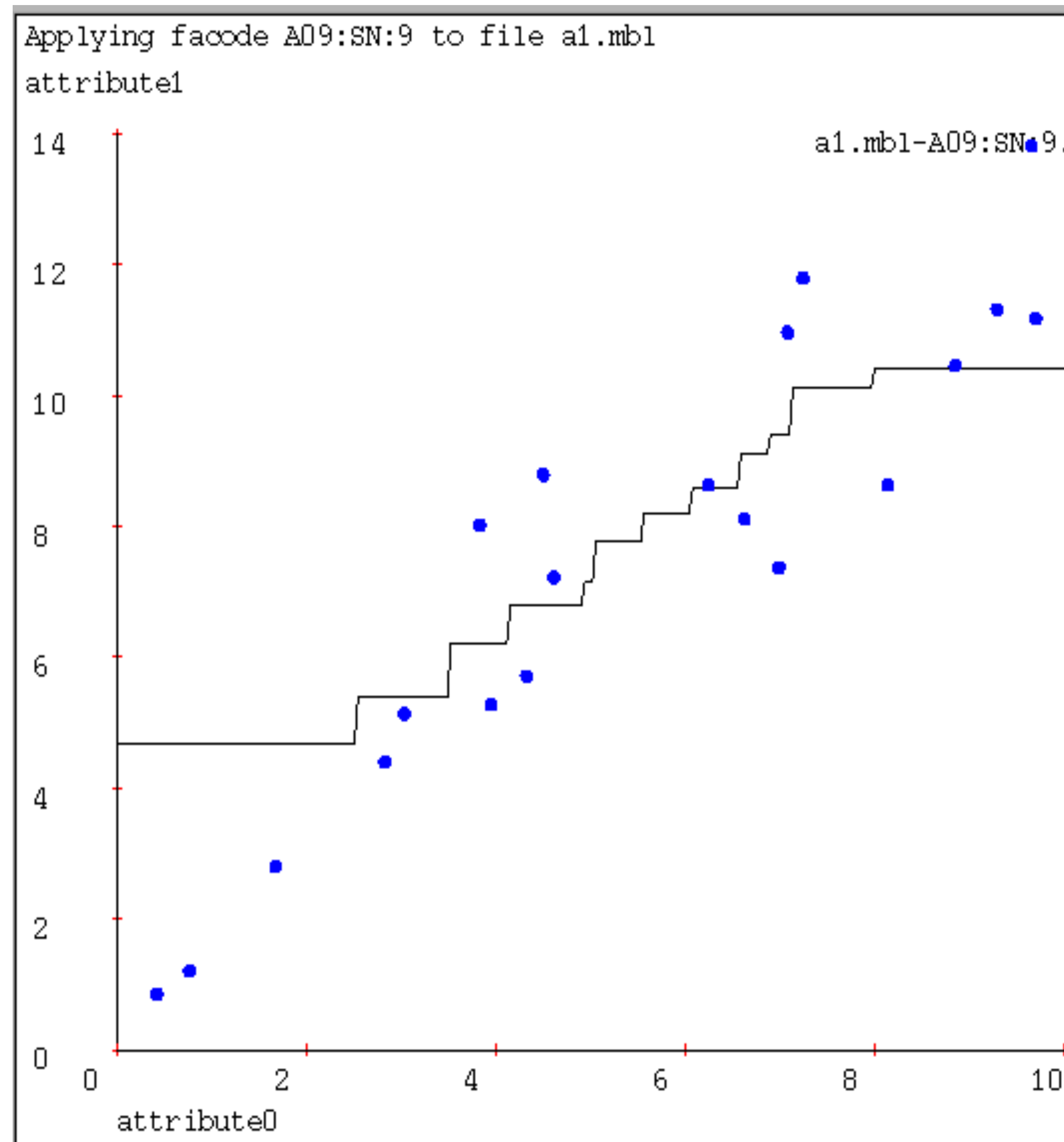


Figure Credit: Andrew Moore

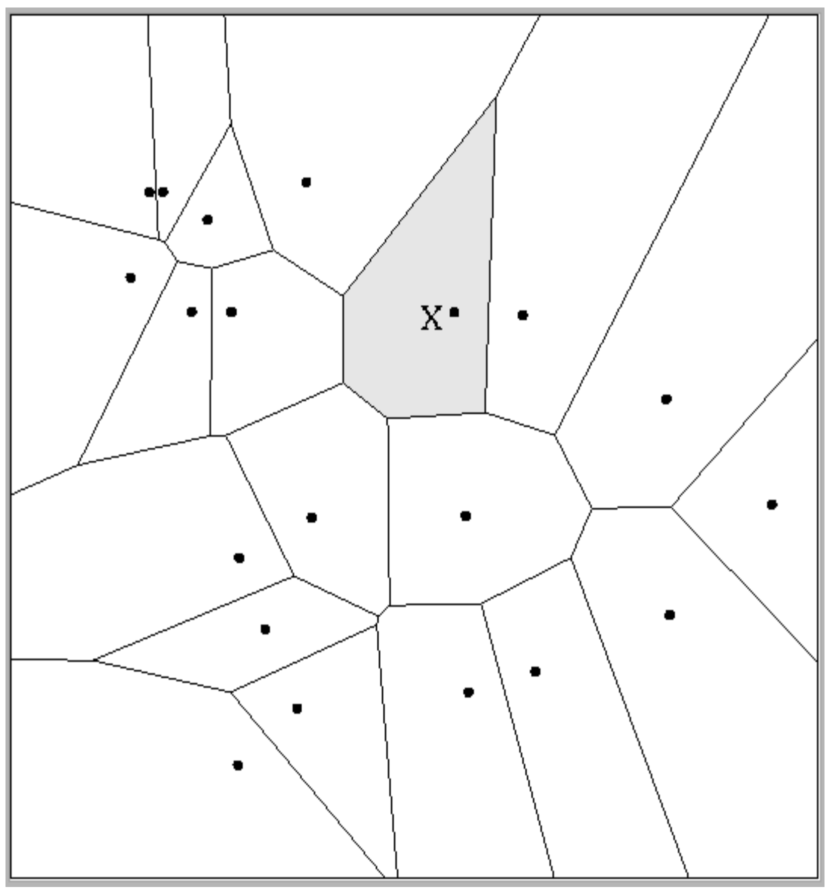
slide by Dhruv Batra

Multivariate distance metrics

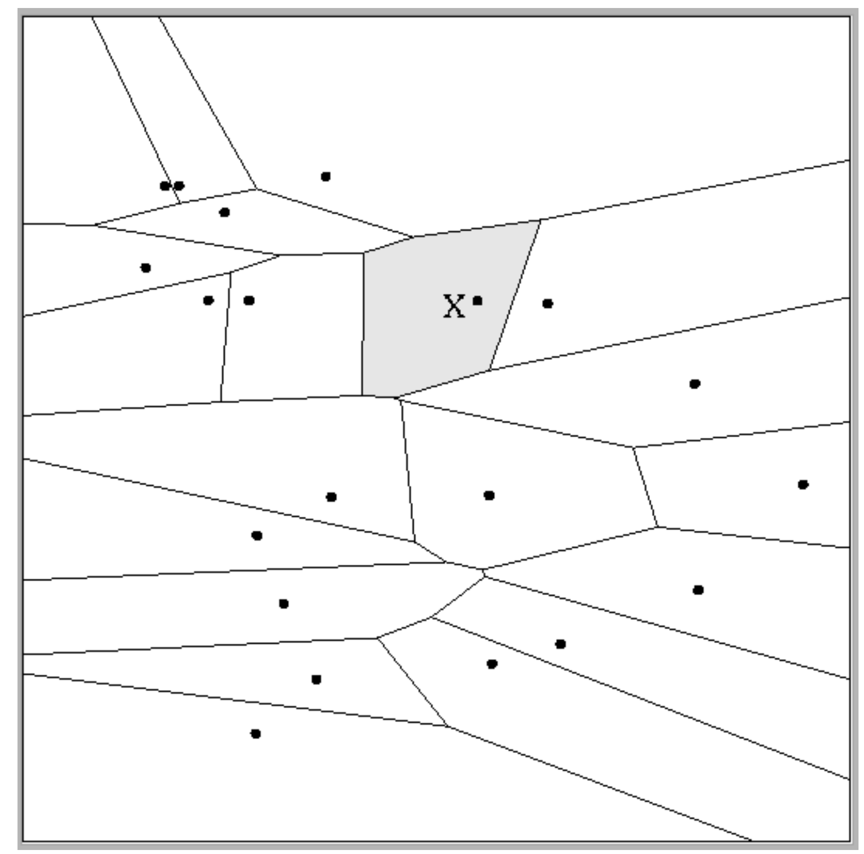
- Suppose the input vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$ are two dimensional:

  $\mathbf{x}_1 = (x_{11}, x_{12}), \mathbf{x}_2 = (x_{21}, x_{22}), \ldots \mathbf{x}_N = (x_{N1}, x_{N2})$.

- One can draw the nearest-neighbor regions in input space.



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



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

The relative scalings in the distance metric affect region shapes

slide by Dhruv Batra

Slide Credit: Carlos Guestrin

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

?

slide by Richard Zemel

Euclidean distance metric

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

Or equivalently,

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

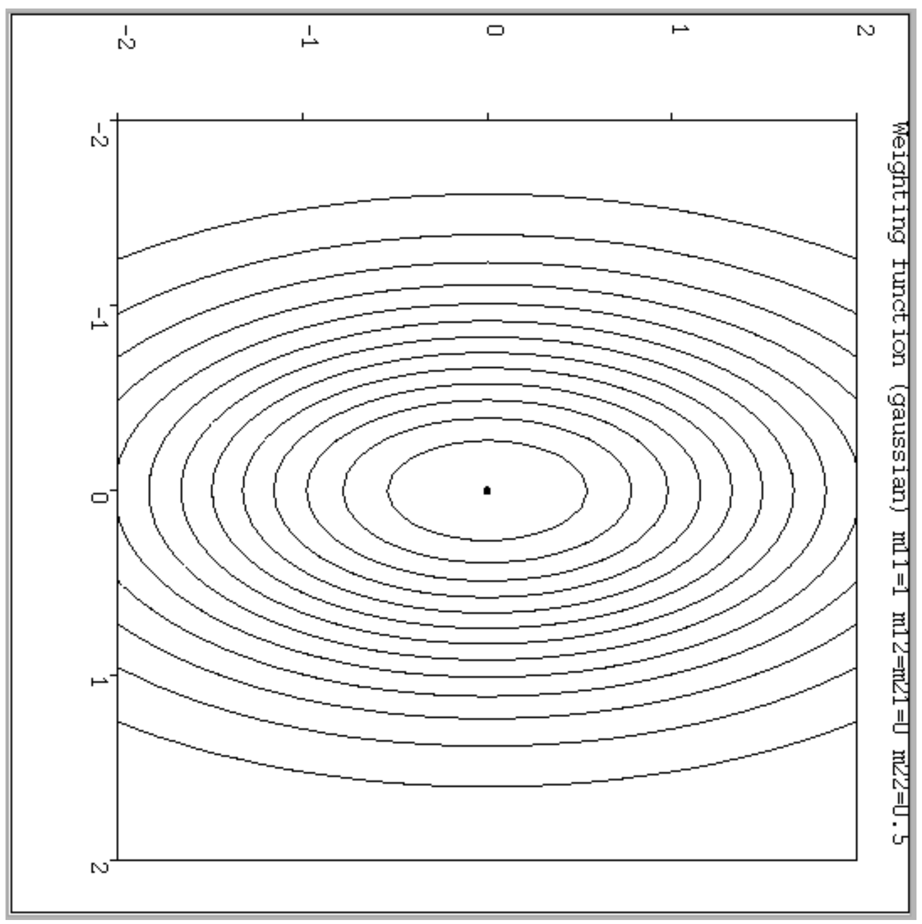
where

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

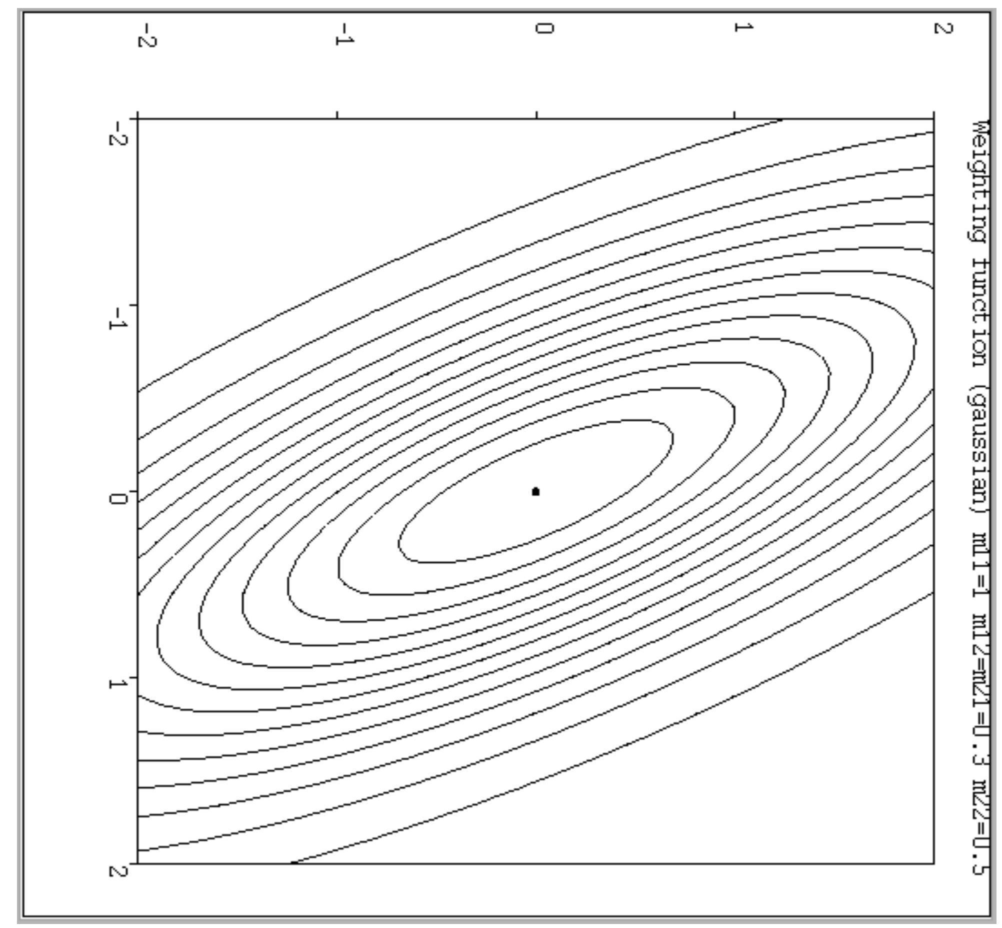
slide by Dhruv Batra

Slide Credit: Carlos Guestrin

Notable distance metrics (and their level sets)



Scaled Euclidian (L_2)

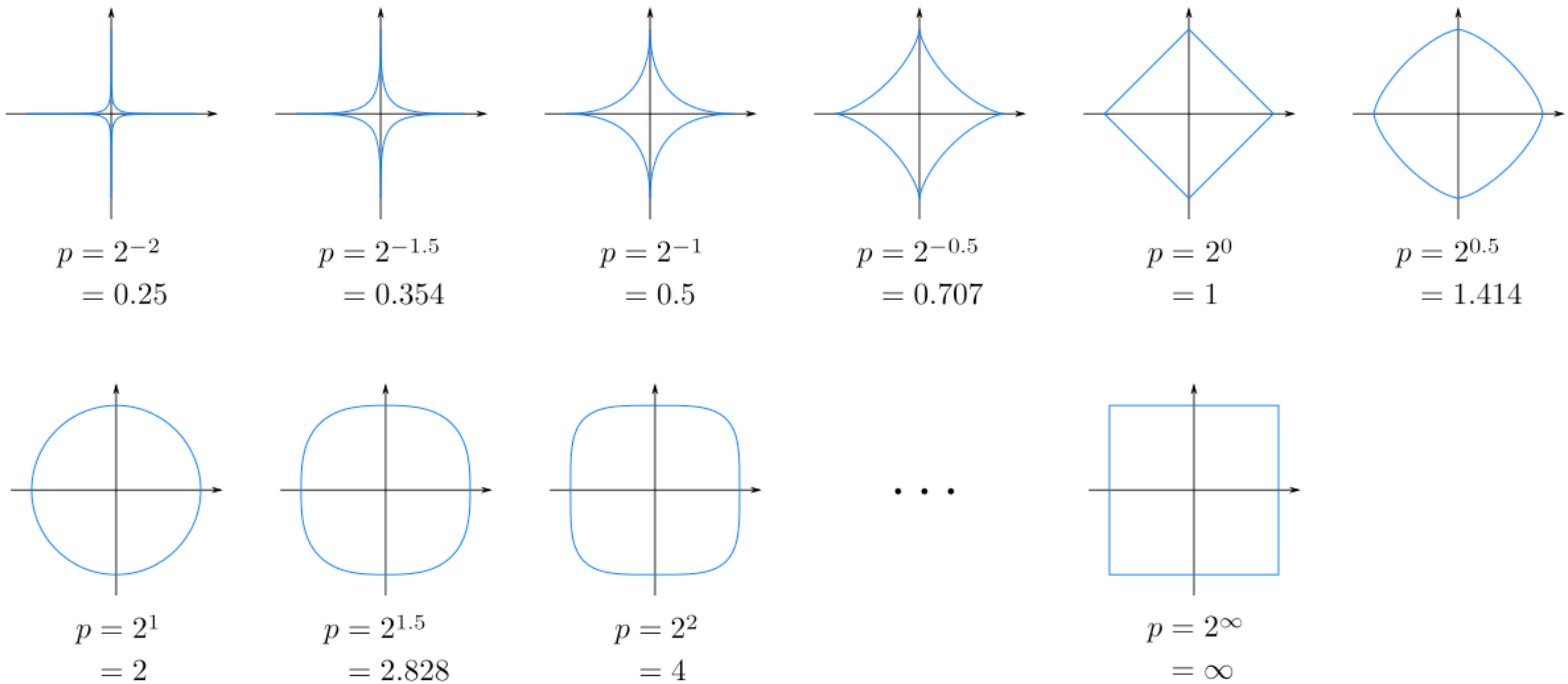


Mahalanobis (non-diagonal A)

slide by Dhruv Batra

Slide Credit: Carlos Guestrin

Minkowski distance

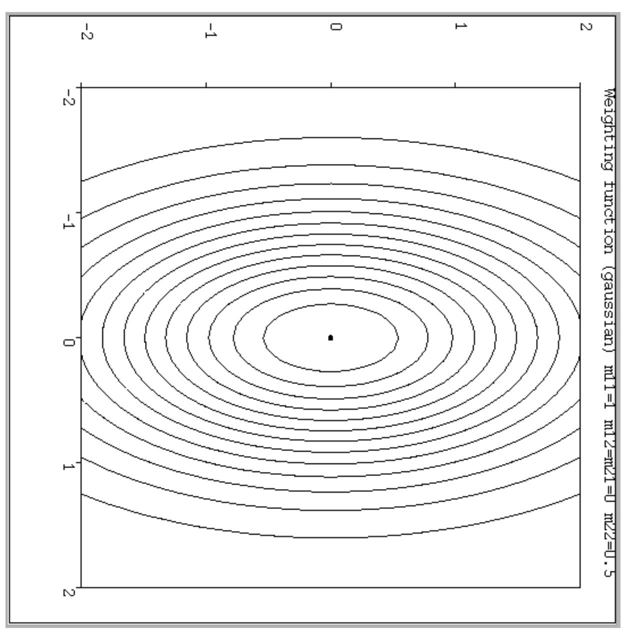


$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

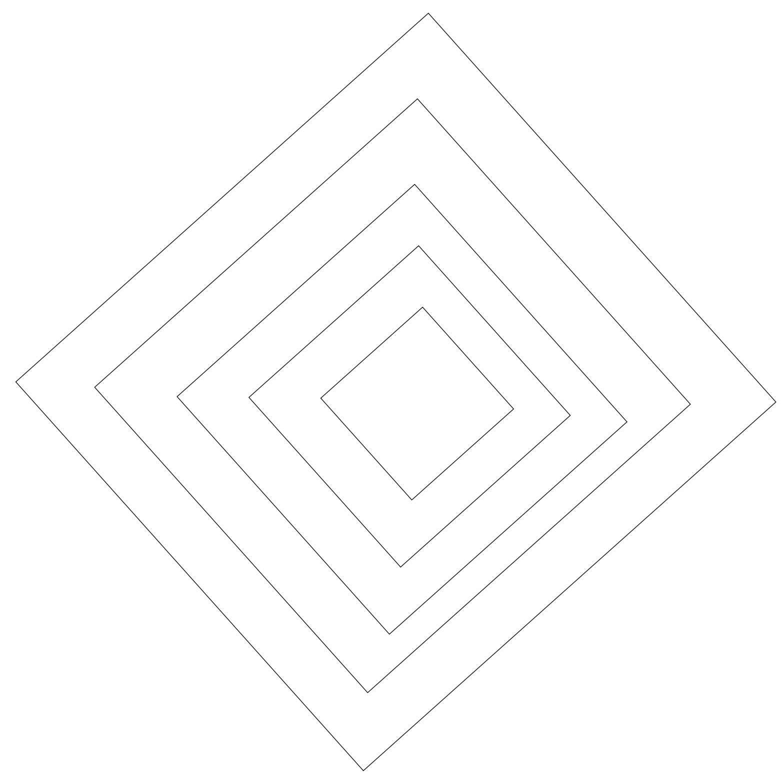
Image Credit: By Waldir (Based on File:MinkowskiCircles.svg)
[CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

slide by Dhruv Batra

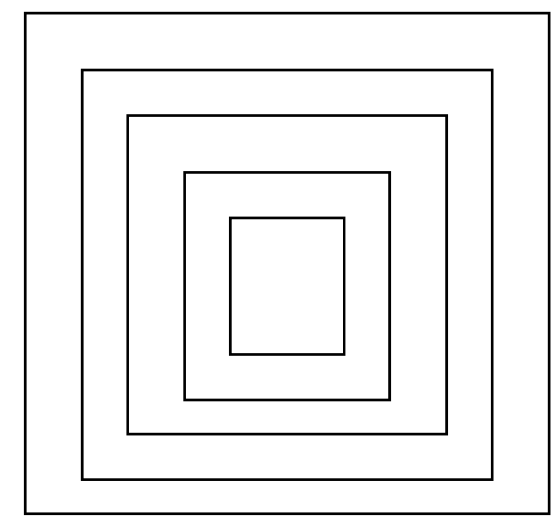
Notable distance metrics (and their level sets)



Scaled Euclidian (L_2)



L_1 norm (absolute)



L_{inf} *(max) norm*

slide by Dhruv Batra

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 $y_i \in \mathcal{R}$
- 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 $K(x_i,x')$

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

slide by Thorsten Joachims

Kernel Regression/Classification

Four things make a memory based learner:

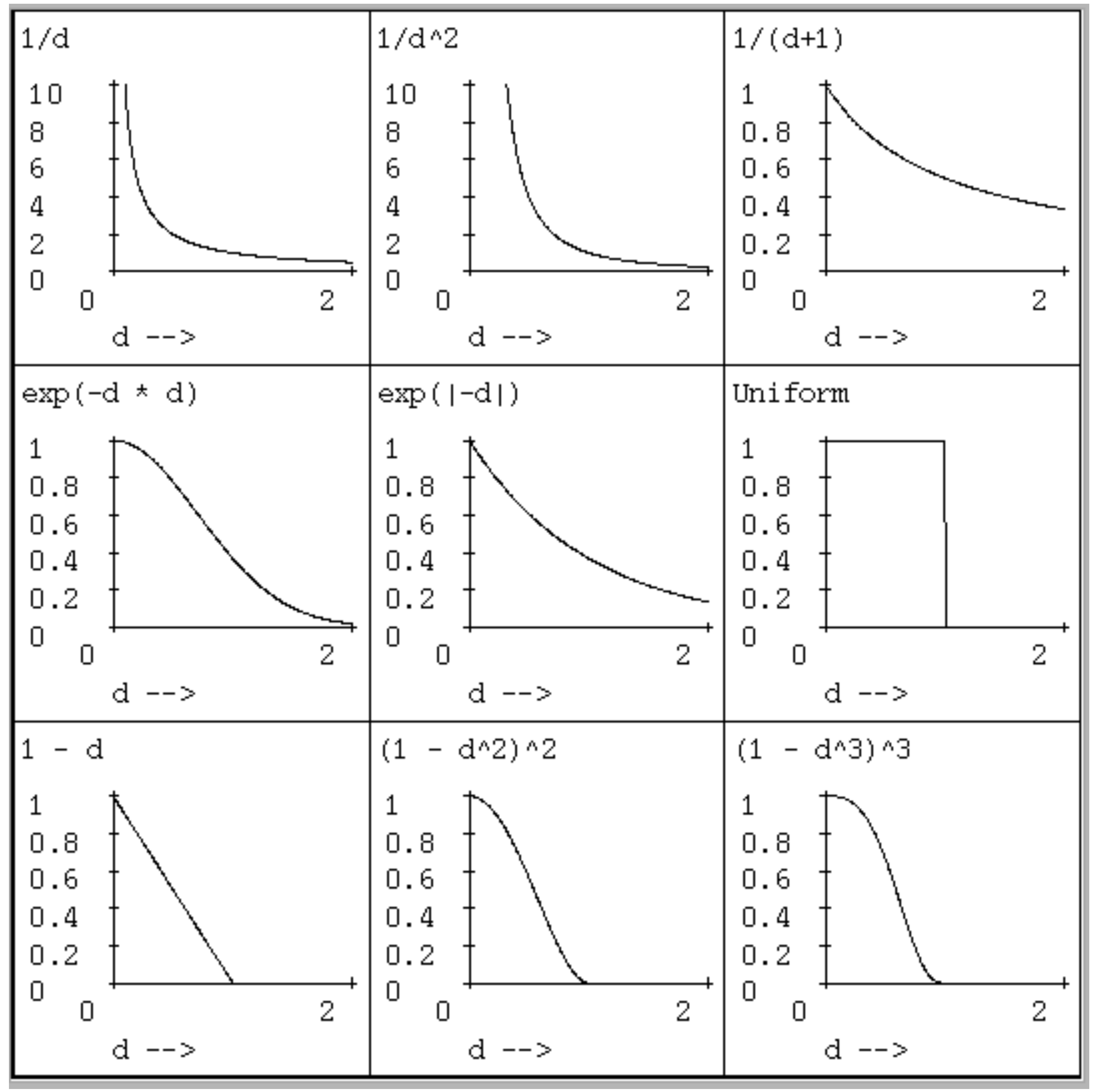
- **A distance metric**
  - Euclidean (and others)
- **How many nearby neighbors to look at?**
  - All of them
- **A weighting function (optional)**
  - $w_i = \exp(-d(x_i, query)^2 / \sigma^2)$
  - Nearby points to the query are weighted strongly, far points weakly. The σ parameter is the Kernel Width.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

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

slide by Dhruv Batra

Effect of Kernel Width

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

- What happens as $\sigma \rightarrow 0$?

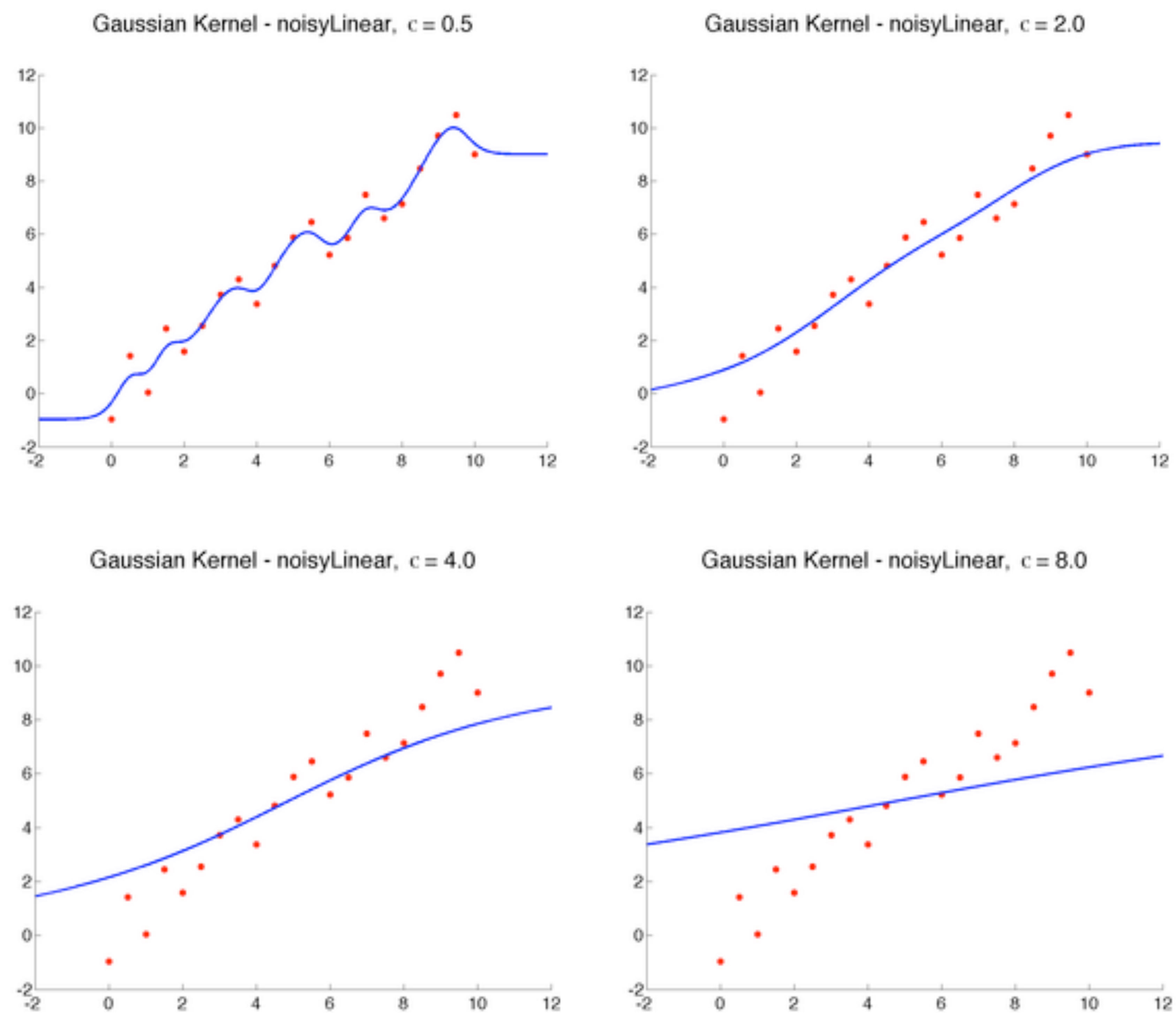


Image Credit: Ben Taskar

slide by Dhruv Batra

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

slide by Dhruv Batra

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

slide by Dhruv Batra

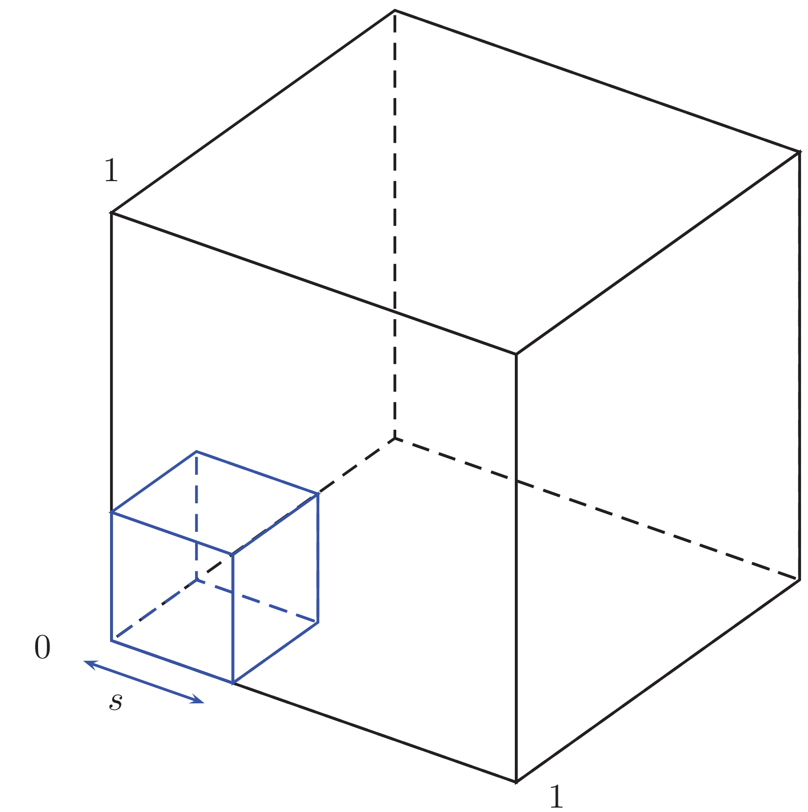
Problems with Instance-Based Learning

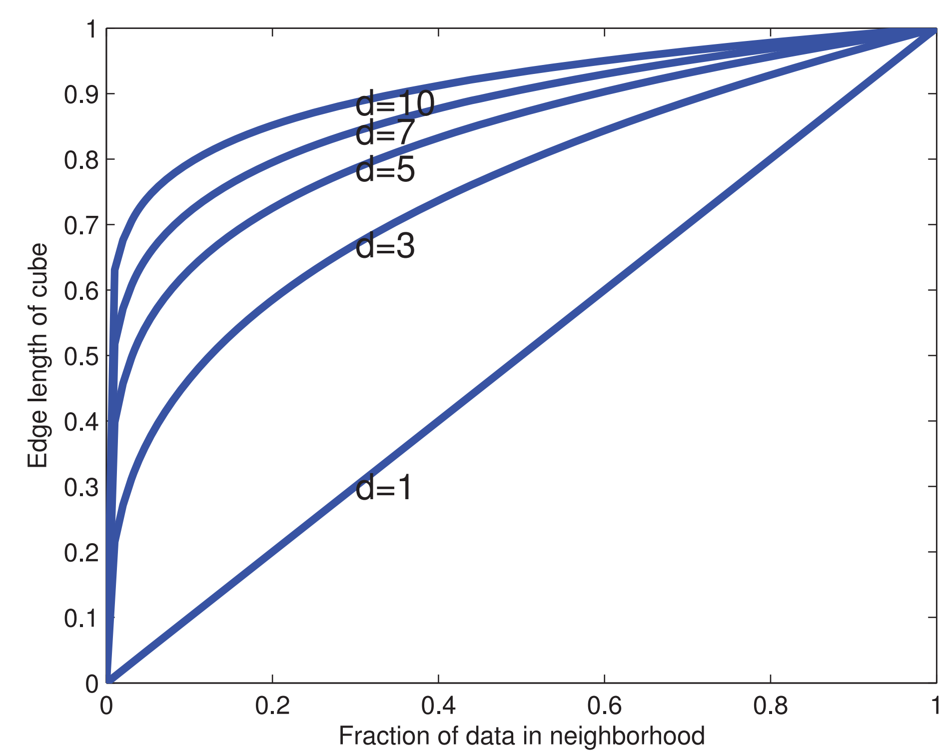
- Expensive
  - No Learning: most real work done during testing
  - For every test sample, must search through all dataset – very slow!
  - Must use tricks like approximate nearest neighbour search
- Doesn't work well when large number of irrelevant features
  - Distances overwhelmed by noisy features
- Curse of Dimensionality
  - Distances become meaningless in high dimensions

slide by Dhruv Batra

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





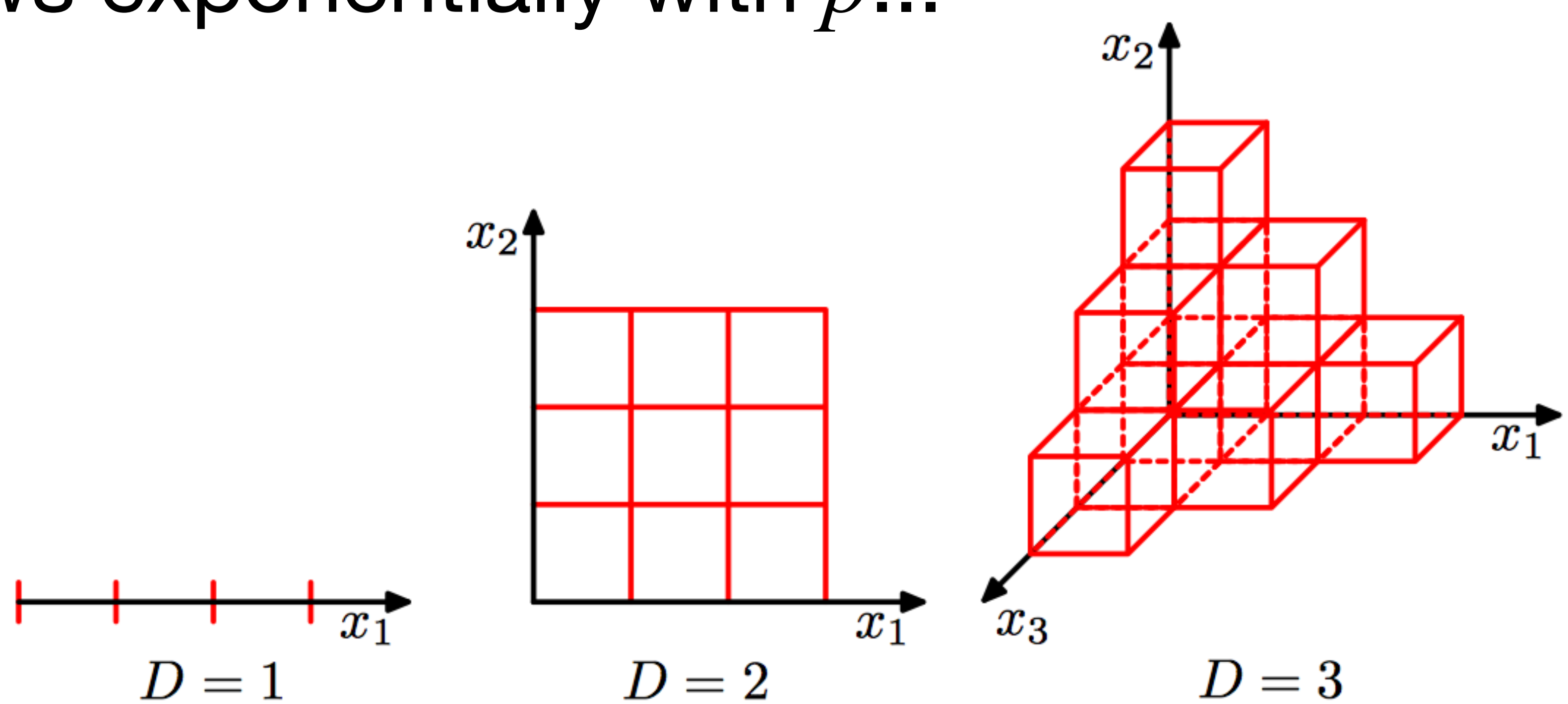
slide by Kevin Murphy

High dimensional spaces are empty

- Assume your data lives in $[0, 1]^p$. The volume of an hypercube with an edge length of $r = 0.1$ is 0.1^p
  → when p grows, it quickly becomes so small that the probability to capture points from your database becomes very very small…

Points in high dimensional spaces are isolated

- To overcome this limitation, you need a number of sample which grows exponentially with p...



slide by Julie Delon

High dimensional spaces are empty

- X, Y two independent variables, with uniform distribution on $[0, 1]^p$. The mean square distance $\|X - Y\|^2$ satisfies

$$\mathbb{E}[\|X - Y\|^2] = p/6 \quad \text{and} \quad Std[\|X - Y\|^2] \simeq 0.2\sqrt{p}$$

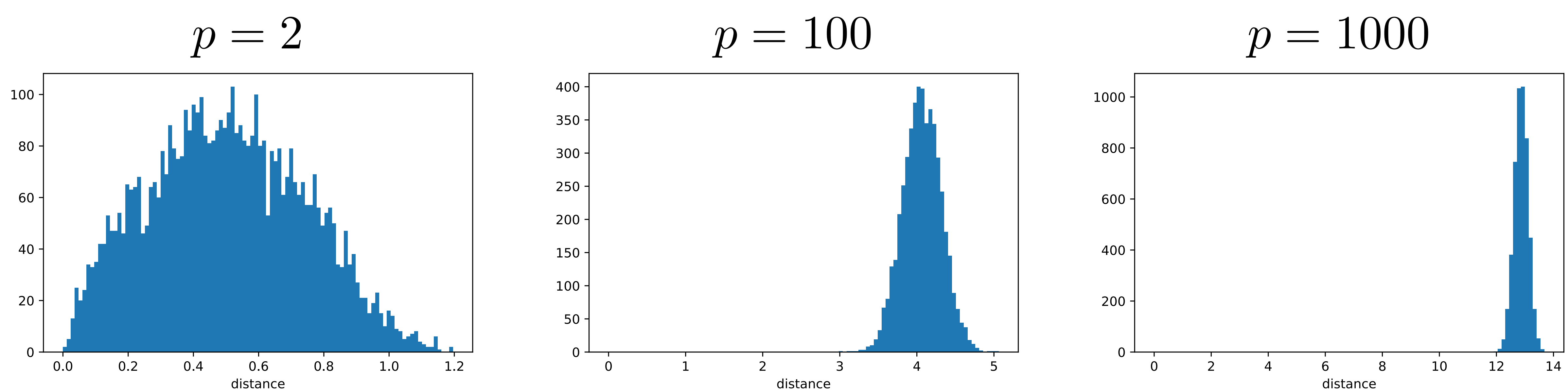


Figure: Histograms of pairwise-distances between $n = 100$ points sampled uniformly in the hypercube $[0, 1]^p$

The notion of nearest neighbors vanishes.

slide by Julie Delon

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

Ways to avoid the curse of dimensionality

- Dimension reduction:
  - the problem comes from that p is too large,
  - therefore, reduce the data dimension to $d \ll p$,
  - such that the curse of dimensionality vanishes!
- Regularization:
  - The problem comes from that parameter estimates are unstable,
  - therefore, regularize these estimates,
  - such that the parameter are correctly estimated!
- Parsimonious models:
  - the problem comes from that the number of parameters to estimate is too large,
  - therefore, make restrictive assumptions on the model,
  - such that the number of parameters to estimate becomes more “decent”!

Next Lecture:

Linear Regression,
Least Squares Optimization,
Model complexity, Regularization