

Instance-based Learning

Nearest Neighbors/Non-Parametric Methods

Machine Learning – CSE546

Carlos Guestrin

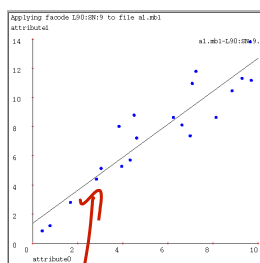
University of Washington

October 21, 2014

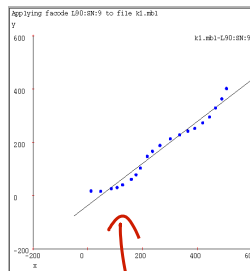
©Carlos Guestrin 2005-2014

1

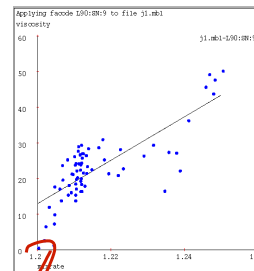
Why not just use Linear Regression?



looks OK



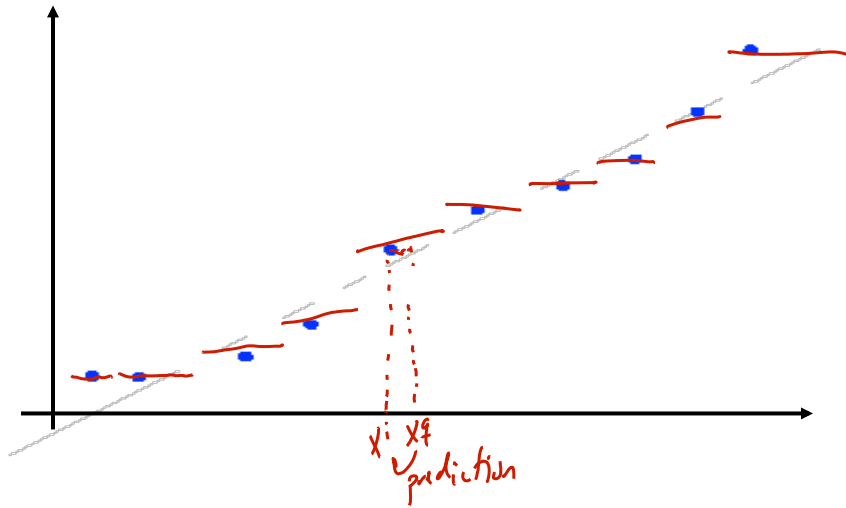
(choose diff. basis functions)



©Carlos Guestrin 2005-2014

2

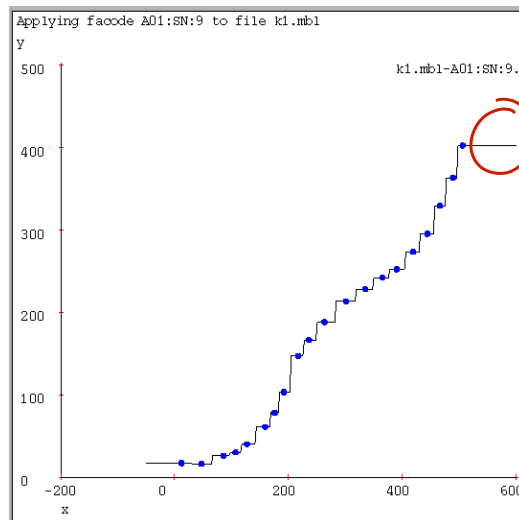
Using data to predict new data



©Carlos Guestrin 2005-2014

3

Nearest neighbor



©Carlos Guestrin 2005-2014

4

Univariate 1-Nearest Neighbor

Given datapoints $(x^1, y^1) (x^2, y^2) \dots (x^N, y^N)$, where we assume $y^i = f(x^i)$ for some unknown function f .

Given query point x^q , your job is to predict $\hat{y} \approx f(x^q)$

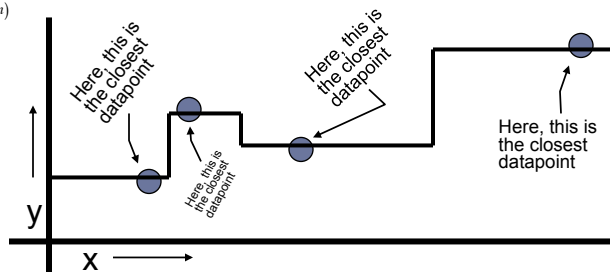
Nearest Neighbor:

1. Find the closest x_i in our set of datapoints

$$j(nn) = \underset{j}{\operatorname{argmin}} |x^j - x^q|$$

2. Predict $\hat{y} = y^{i(nn)}$

Here's a dataset with one input, one output and four datapoints.



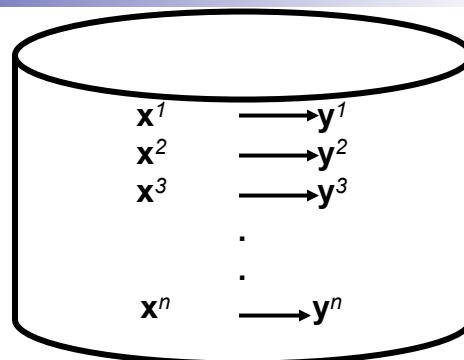
©Carlos Guestrin 2005-2014

5

1-Nearest Neighbor is an example of.... Instance-based learning

A function approximator that has been around since about 1910.

To make a prediction, search database for similar datapoints, and fit with the local points.



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?

©Carlos Guestrin 2005-2014

6

1-Nearest Neighbor

Four things make a memory based learner:

1. A distance metric
Euclidian (and many more)
2. How many nearby neighbors to look at?
One
3. A weighting function (optional)
Unused
4. How to fit with the local points?
Just predict the same output as the nearest neighbor.

$$i = \operatorname{argmin}_j \|x^j - x^k\|$$

predict $\hat{y} = y^i$

©Carlos Guestrin 2005-2014

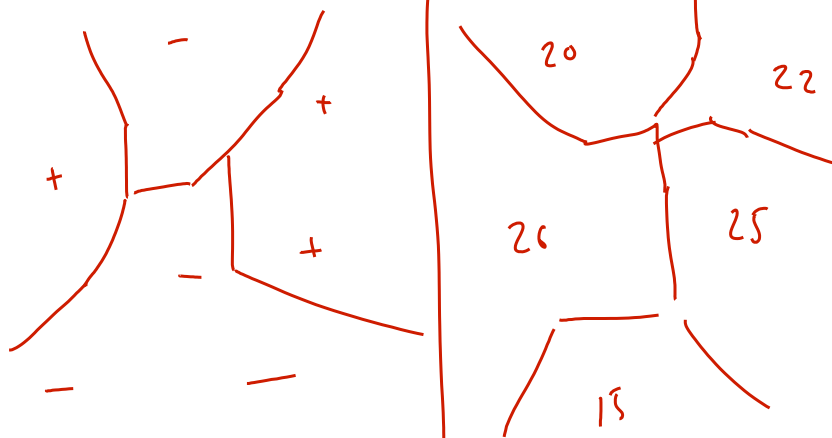
7

Multivariate 1-NN examples

Classification

Voronoi diagram

Regression



©Carlos Guestrin 2005-2014

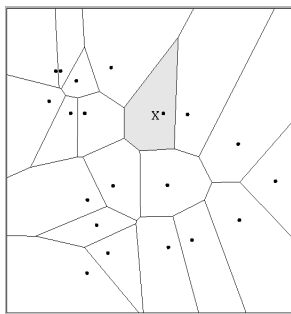
8

Multivariate distance metrics

Suppose the input vectors x^1, x^2, \dots, x^N are two dimensional:

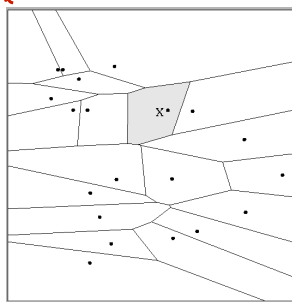
$$\mathbf{x}^1 = (x^1_1, x^1_2), \mathbf{x}^2 = (x^2_1, x^2_2), \dots, \mathbf{x}^N = (x^N_1, x^N_2).$$

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



$$Dist(\mathbf{x}^i, \mathbf{x}^j) = (x^i_1 - x^j_1)^2 + (x^i_2 - x^j_2)^2$$

care mark about variations in x_2



$$Dist(\mathbf{x}^i, \mathbf{x}^j) = (x^i_1 - x^j_1)^2 + (3x^i_2 - 3x^j_2)^2$$

The relative scalings in the distance metric affect region shapes

©Carlos Guestrin 2005-2014

9

Euclidean distance metric

Or equivalently,

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

where

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

choice of Σ has a huge impact:

Learning distance metrics

↳ learning params of Σ

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

change space by general Σ

or just weigh each dim

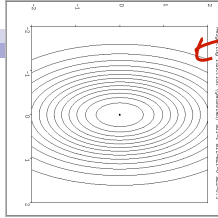
Other Metrics...

- Mahalanobis, Rank-based, Correlation-based,...

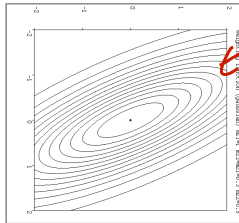
©Carlos Guestrin 2005-2014

10

Notable distance metrics (and their level sets)



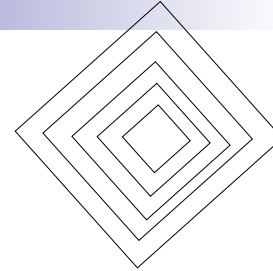
Scaled Euclidian (L_2)



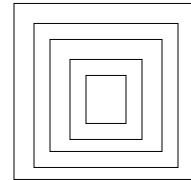
Mahalanobis (here, Σ on the previous slide is not necessarily diagonal, but is symmetric)

$\Sigma \leftarrow$ diagonal

\leftarrow general Σ



L_1 norm (absolute)



$L1$ (max) norm

©Carlos Guestrin 2005-2014

11

Consistency of 1-NN

- Consider an estimator f_n trained on n examples
 - e.g., 1-NN, neural nets, regression,...
- Estimator is *consistent* if true error goes to zero as amount of data increases
 - e.g., for no noise data, consistent if:

$$\lim_{n \rightarrow \infty} \overset{\text{Error bias}}{MSE}(f_n) = 0$$

- Regression is not consistent!
 - Representation bias
- 1-NN is consistent** (under some mild fineprint)



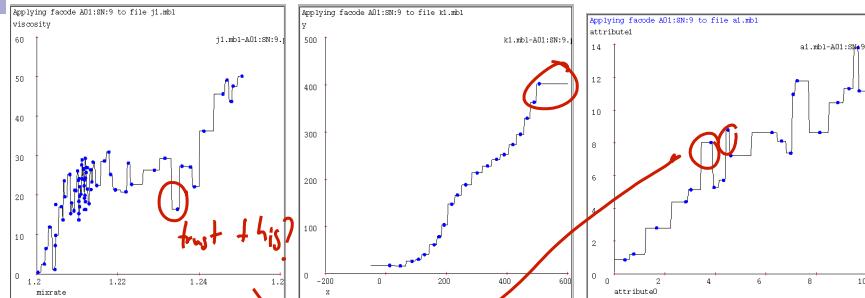
What about variance???

can be really high:
finite sample error can be high

©Carlos Guestrin 2005-2014

12

1-NN overfits?



1-NN has too much flexibility. Want a smoother function.

©Carlos Guestrin 2005-2014

13

k-Nearest Neighbor

Four things make a memory based learner:

1. A distance metric
Euclidian (and many more)
2. How many nearby neighbors to look at?
k
1. A weighting function (optional)
Unused
2. How to fit with the local points?
Just predict the average output among the k nearest neighbors.

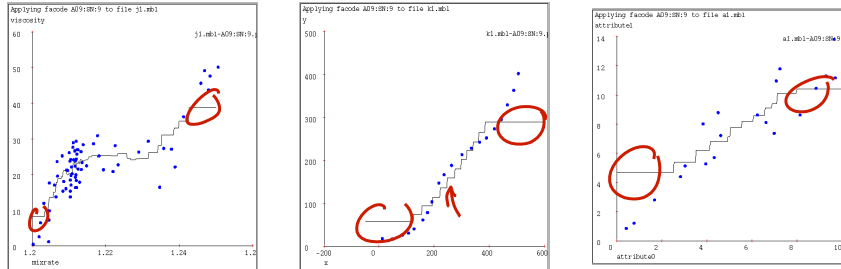
$NN(x^q) \leftarrow k$ nearest neighbors

in regression: $\hat{y} = \frac{1}{k} \sum_{i \in NN(x^q)} y_i$ | in classification: majority vote

©Carlos Guestrin 2005-2014

14

k-Nearest Neighbor (here k=9)



K-nearest neighbor for function fitting smoothes away noise, but there are clear deficiencies.

What can we do about all the discontinuities that k-NN gives us?

©Carlos Guestrin 2005-2014

15

Weighted k-NNs

- Neighbors are not all the same

$x^1 \dots x^k$
 x^2
 x^3

$$\hat{y} = \frac{\pi_1 y^1 + \pi_2 y^2 + \pi_3 y^3}{\pi_1 + \pi_2 + \pi_3}$$

π_i is some weight

e.g., $\pi_i = \frac{1}{\|x^k - x^i\|}$

©Carlos Guestrin 2005-2014

16

Kernel regression

Four things make a memory based learner:

1. A distance metric
Euclidian (and many more)
2. How many nearby neighbors to look at?
All of them
3. A weighting function (optional)
 $\pi^i = \exp(-D(x^i, query)^2 / \rho^2)$

Nearby points to the query are weighted strongly, far points weakly. The ρ parameter is the **Kernel Width**. Very important.

4. How to fit with the local points?

Predict the weighted average of the outputs:

$$\text{predict} = \frac{\sum \pi^i y^i}{\sum \pi^i}$$

Regression

Classification
weighted voted

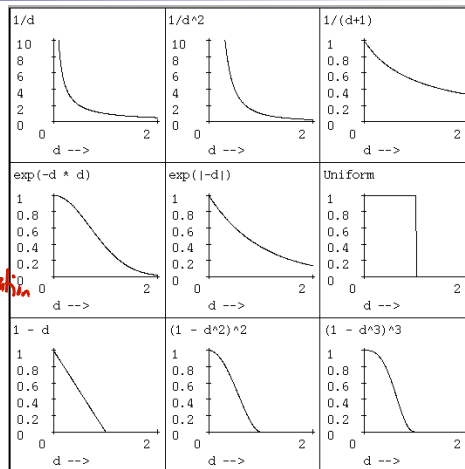
Kernel

$$\pi(x^i, x^q) = e^{-\frac{\|x^i - x^q\|^2}{\rho^2}}$$

Weighting functions

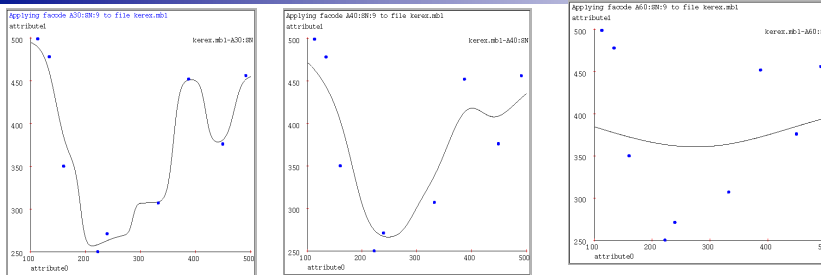
$$\pi^i = \exp(-D(x^i, query)^2 / \rho^2)$$

tune ρ
bandwidth
with cross validation



Typically optimize ρ using \leftarrow not a convex gradient descent (Our examples use Gaussian)

Kernel regression predictions



$\rho=10$

less smooth

$\rho=20$

just right

$\rho=80$

too smooth

Increasing the kernel width ρ means further away points get an opportunity to influence you.

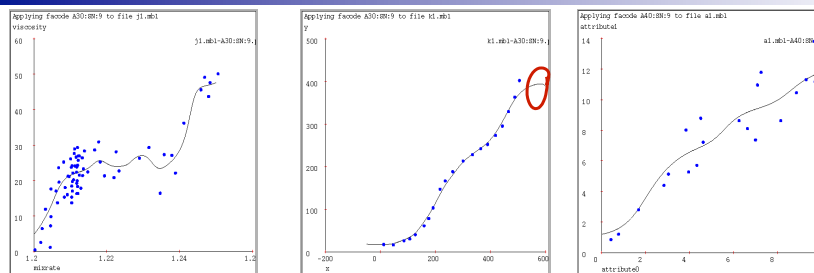
As $\rho \rightarrow \infty$, the prediction tends to the global average.

$\rho \rightarrow \infty$
 $\rightarrow 1\text{-NN}$

©Carlos Guestrin 2005-2014

19

Kernel regression on our test cases



$\rho=1/32$ of x-axis width.

$\rho=1/32$ of x-axis width.

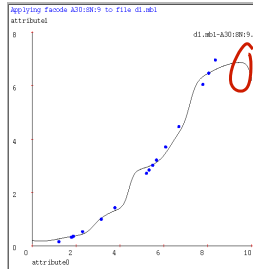
$\rho=1/16$ axis width.

Choosing a good ρ is important. Not just for Kernel Regression, but for all the locally weighted learners we're about to see.

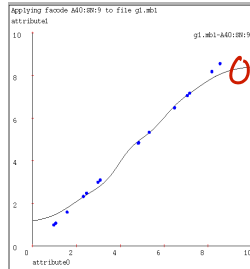
©Carlos Guestrin 2005-2014

20

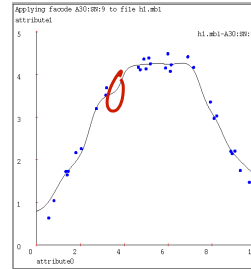
Kernel regression can look bad



$\rho = \text{Best.}$



$\rho = \text{Best.}$



$\rho = \text{Best.}$

Time to try something more powerful...

©Carlos Guestrin 2005-2014

21

Locally weighted regression

Kernel regression:

Take a very very conservative function approximator called AVERAGING. Locally weight it.

Locally weighted regression:

Take a conservative function approximator called LINEAR REGRESSION. Locally weight it.

©Carlos Guestrin 2005-2014

22

Locally weighted regression

- Four things make a memory based learner:

- A distance metric

Any

- How many nearby neighbors to look at?

All of them

- A weighting function (optional)

Kernels

$$\pi^i = \exp(-D(x^i, \text{query})^2 / \rho^2)$$

- How to fit with the local points?

General weighted regression:

$$\hat{g} = \hat{w} \cdot x^q$$

$$\hat{w}^q = \underset{w}{\operatorname{argmin}} \sum_{k=1}^N \pi_q^k (y^k - w^T x^k)^2$$

fit a weighted linear model

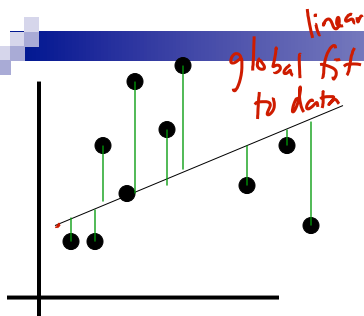
weights

e.g. squared error

©Carlos Guestrin 2005-2014

23

How LWR works

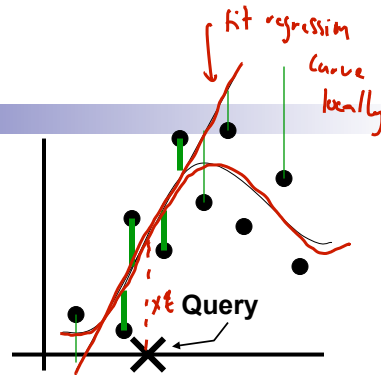


Linear regression

- Same parameters for all queries

$$\hat{w} = (X^T X)^{-1} X^T Y$$

weighted version of



Locally weighted regression

- Solve weighted linear regression for each query

$$w^q = ((\Pi X)^T \Pi X)^{-1} (\Pi X)^T \Pi Y$$

$$\Pi = \begin{pmatrix} \pi_1 & 0 & 0 & 0 \\ 0 & \pi_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \pi_n \end{pmatrix}$$

©Carlos Guestrin 2005-2014

24

Another view of LWR

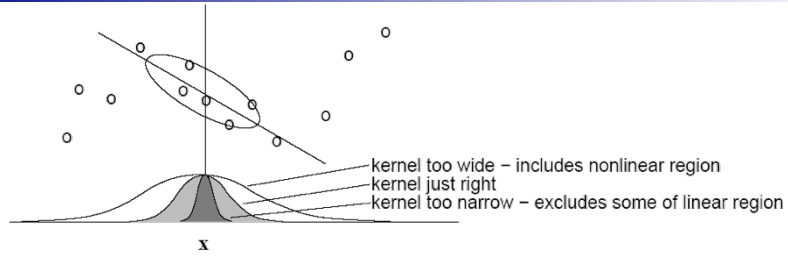
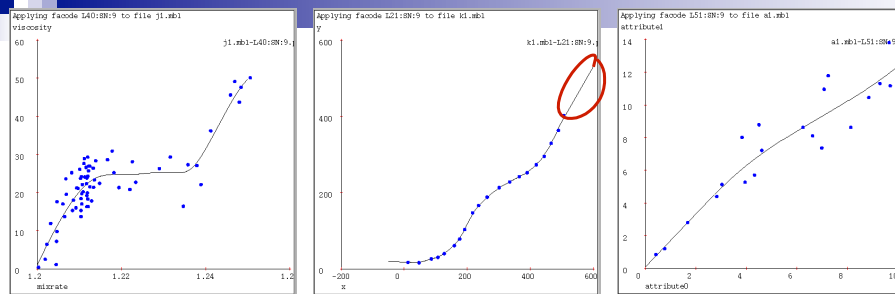


Image from Cohn, D.A., Ghahramani, Z., and Jordan, M.I. (1996) "Active Learning with Statistical Models", JAIR Volume 4, pages 129-145.

LWR on our test cases



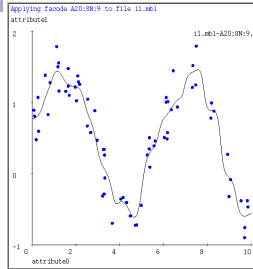
$\rho = 1/16$ of x-axis width.

$\rho = 1/32$ of x-axis width.

$\rho = 1/8$ of x-axis width.

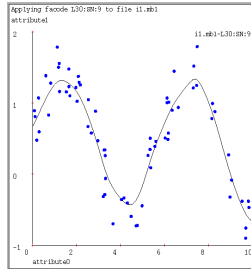
Locally weighted polynomial regression

Semi parametric model: some parameters, a regression, some flexibility \rightarrow NN, costly



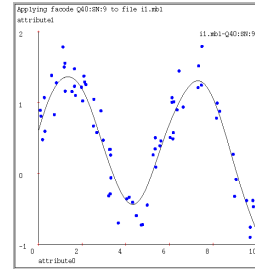
Kernel Regression
Kernel width ρ at optimal level.

$\rho = 1/100$ x-axis



LW Linear Regression
Kernel width ρ at optimal level.

$\rho = 1/40$ x-axis



LW Quadratic Regression
Kernel width ρ at optimal level.

$\rho = 1/15$ x-axis

→ smoother

Local quadratic regression is easy: just add quadratic terms to the X matrix. As the regression degree increases, the kernel width can increase without introducing bias.

©Carlos Guestrin 2005-2014

27

Curse of dimensionality for instance-based learning

- Must store and retrieve all data!
 - Most real work done during testing
 - For every test sample, must search through all dataset – very slow!
 - There are (sometimes) fast methods for dealing with large datasets
- Instance-based learning often poor with noisy or irrelevant features

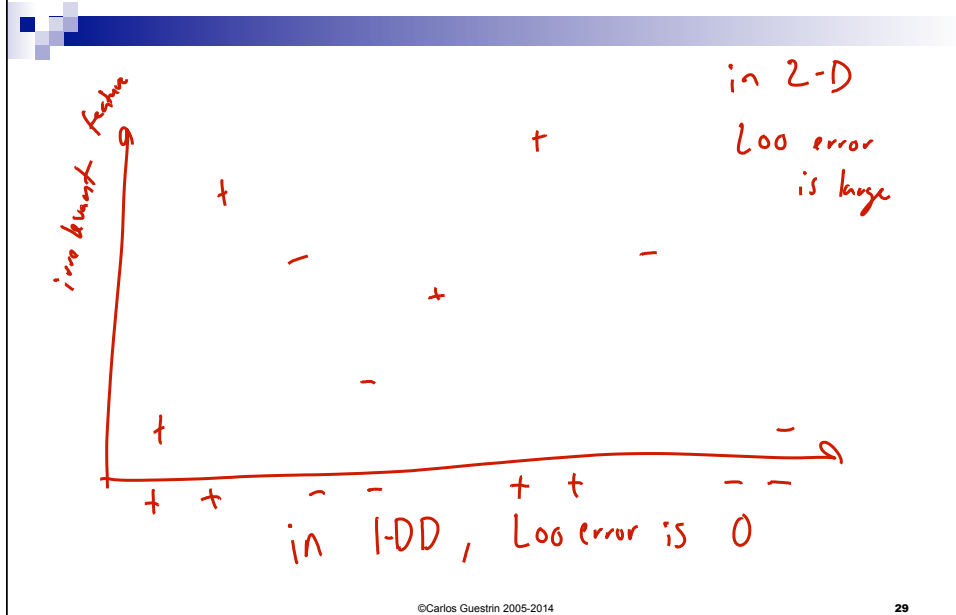
error rate (NN) $\rightarrow 0$

\rightarrow rate exponentially worse with dim of data

©Carlos Guestrin 2005-2014

28

Curse of the irrelevant feature



What you need to know about instance-based learning

- k-NN
 - Simplest learning algorithm
 - With sufficient data, very hard to beat “strawman” approach
 - Picking k?
- Kernel regression
 - Set k to n (number of data points) and optimize weights by gradient descent
 - Smoother than k-NN
- Locally weighted regression
 - Generalizes kernel regression, not just local average
- Curse of dimensionality
 - Must remember (very large) dataset for prediction
 - Irrelevant features often killers for instance-based approaches

Acknowledgment

- This lecture contains some material from Andrew Moore's excellent collection of ML tutorials:
 - <http://www.cs.cmu.edu/~awm/tutorials>