Subhransu Maji

CMPSCI 689: Machine Learning

29 January 2015

3 February 2015

Topics of interest

hw00 poll

- NLP 13
- Deep learning, neural networks 8
- ◆ Computer vision 8
- Information retrieval 8
- ◆ Databases, systems, networking 4
- ◆ Al 3
- ◆ Reinforcement learning 3
- ◆ Robotics 3
- ◆ These got 1 or 2 mentions:
 - complexity, logic, large scale learning, speech, cross modality, biology, neuroscience, graphics, recommender systems, semisupervised learning, programming languages, virtual reality, privacy, security

"To pass the class with a B+"

Topics of interest

hw00 poll

- ◆ NLP 13
- ◆ Deep learning, neural networks 8
- ◆ Computer vision 8
- ◆ Information retrieval 8
- Databases, systems, networking 4
- ◆ Al 3
- ◆ Reinforcement learning 3
- ◆ Robotics 3
- ◆ These got 1 or 2 mentions:
 - complexity, logic, large scale learning, speech, cross modality, biology, neuroscience, graphics, recommender systems, semisupervised learning, programming languages, virtual reality, privacy, security

CMPSCI 689

Subhransu Maji (UMASS)

2/37

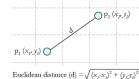
Nearest neighbor classifier

CMPSCI 689 Subhransu Maji (UMASS) 2/37 CMPSCI 689 Subhransu Maji (UMASS) 3/37

- Will Alice like AI?
 - Alice and James are similar and James likes Al. Hence, Alice must also like Al.

Nearest neighbor classifier

- ◆ Will Alice like AI?
 - Alice and James are similar and James likes Al. Hence, Alice must also like Al.
- It is useful to think of data as feature vectors
 - Use Euclidean distance to measure similarity



CMPSCI 689

Subhransu Maji (UMASS)

3/37

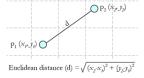
CMPSCI 689

Subhransu Maji (UMASS)

3/37

Nearest neighbor classifier

- ♦ Will Alice like Al?
- Alice and James are similar and James likes Al. Hence, Alice must also like Al.
- It is useful to think of data as feature vectors
 - Use Euclidean distance to measure similarity
- Data to feature vectors



Nearest neighbor classifier

- ◆ Will Alice like AI?
 - Alice and James are similar and James likes Al. Hence, Alice must also like Al.
- ◆ It is useful to think of data as feature vectors
 - Use Euclidean distance to measure similarity
- ◆ Data to feature vectors
 - ▶ Binary: e.g. Al? {no, yes}
 - **→** {0,1}
 - → or {-20, 2}

 $p_1\left(x_p,y_p\right)$ Euclidean distance (d) = $\sqrt{(x_2,x_j)^2 + (y_2,y_p)^2}$

 CMPSCI 689
 Subhransu Maji (UMASS)
 3/37
 CMPSCI 689
 Subhransu Maji (UMASS)
 3/37

◆ Will Alice like AI?

Alice and James are similar and James likes Al. Hence, Alice must also like Al.

It is useful to think of data as feature vectors

Use Euclidean distance to measure similarity

Data to feature vectors

▶ Binary: e.g. Al? {no, yes}

- **→** {0,1}
- **→** or {-20, 2} **X**

Nearest neighbor classifier

♦ Will Alice like AI?

Alice and James are similar and James likes Al. Hence, Alice must also like Al.

- It is useful to think of data as feature vectors
 - Use Euclidean distance to measure similarity
- ◆ Data to feature vectors

▶ Binary: e.g. Al? {no, yes}

- **→** {0,1}
- → or {-20, 2} X

▶ Nominal: e.g. color = {red, blue, green, yellow}

- → {0,1}ⁿ
- **→** or {0,1,2,3}

CMPSCI 689 Subhransu Maji (UMASS) 3/37 CMPSCI 689 Subhransu Maji (UMASS)

 $\bigcap P_2 (x_2, y_2)$

3/37

 $\bigcap P_2(x_2, y_2)$

 $p_1(x_p, y_p)$

 $p_1(x_0,y_i)$

Euclidean distance (d) = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

Euclidean distance (d) = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

Nearest neighbor classifier

♦ Will Alice like AI?

Alice and James are similar and James likes Al. Hence, Alice must also like Al.

It is useful to think of data as feature vectors

Use Euclidean distance to measure similarity

Data to feature vectors

• Binary: e.g. Al? {no, yes}

- **→** {0,1}
- → or {-20, 2} X

Nominal: e.g. color = {red, blue, green, yellow}

- \rightarrow {0,1}ⁿ
- → or {0,1,2,3} **X**

Nearest neighbor classifier

◆ Will Alice like AI?

Alice and James are similar and James likes Al. Hence, Alice must also like Al.

Subhransu Maji (UMASS)

◆ It is useful to think of data as feature vectors

Use Euclidean distance to measure similarity

◆ Data to feature vectors

• Binary: e.g. Al? {no, yes}

- **→** {0,1}
- → or {-20, 2} X

▶ Nominal: e.g. color = {red, blue, green, yellow}

- \rightarrow {0,1}ⁿ
- **→** or {0,1,2,3} **X**

Real valued: e.g. temperature

→ copied

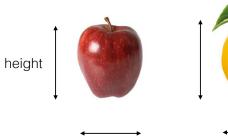
CMPSCI 689

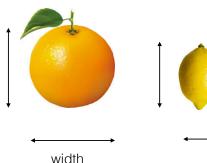
→ or {low, medium, high}

 $p_1(x_p, y_t) \bigcirc$ Euclidean distance (d) = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$ 3/37 $\bigcap P_2 (x_2, y_2)$ $p_1(x_p, y_p)$ Euclidean distance (d) = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

P2 (x2 y2)

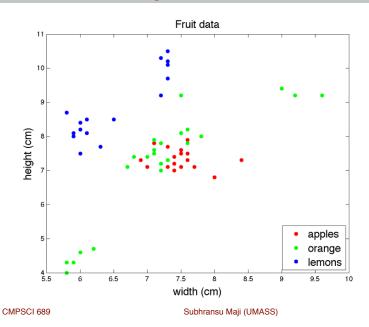
- ullet Training data is in the form of $(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_n,y_n)$
- ◆ Fruit data:
 - ▶ label: {apples, oranges, lemons}
- attributes: {width, height}
- ullet Euclidean distance $d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_i \left(\mathbf{x}_{1,i} \mathbf{x}_{2,i}\right)^2}$



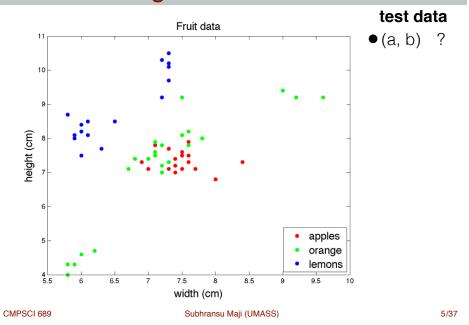


CMPSCI 689 Subhransu Maji (UMASS) 4/37

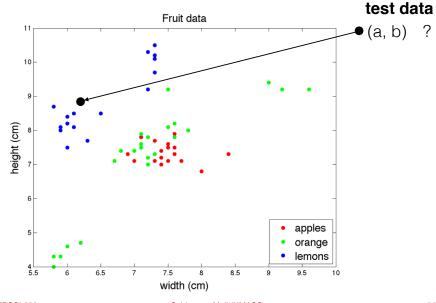
Nearest neighbor classifier



Nearest neighbor classifier

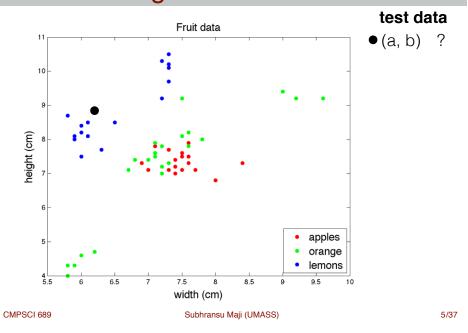


Nearest neighbor classifier

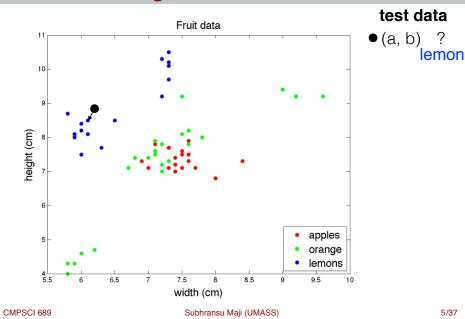


5/37

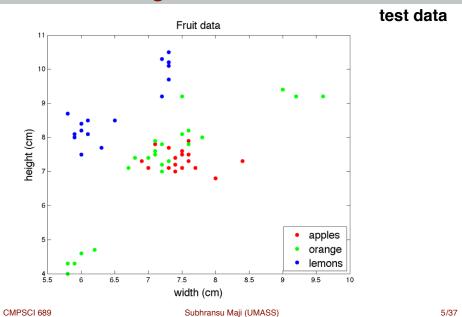
CMPSCI 689 Subhransu Maji (UMASS) 5/37



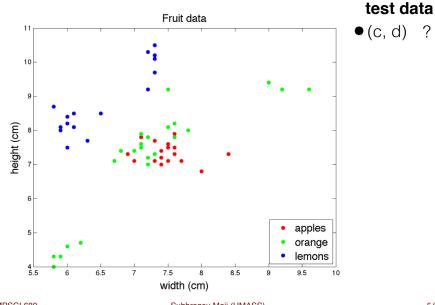
Nearest neighbor classifier



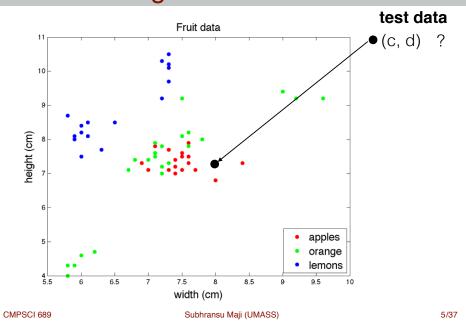
Nearest neighbor classifier



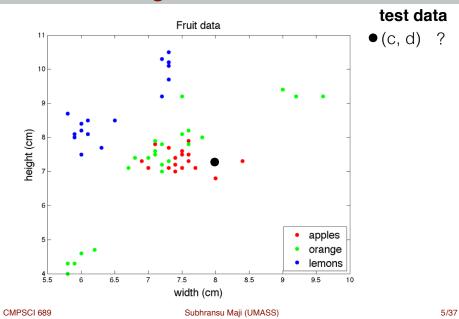
Nearest neighbor classifier



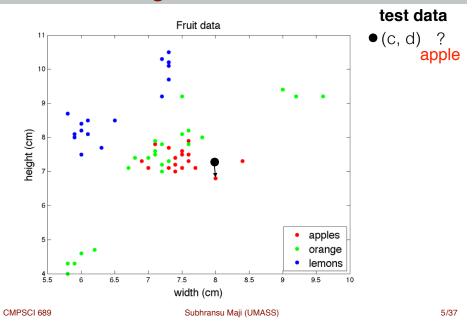
CMPSCI 689 Subhransu Maji (UMASS) 5/37



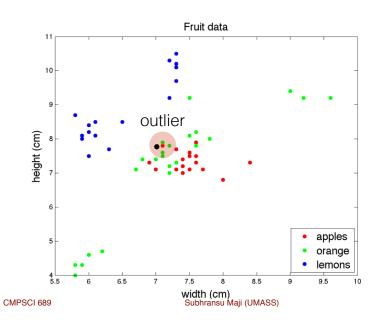
Nearest neighbor classifier



Nearest neighbor classifier

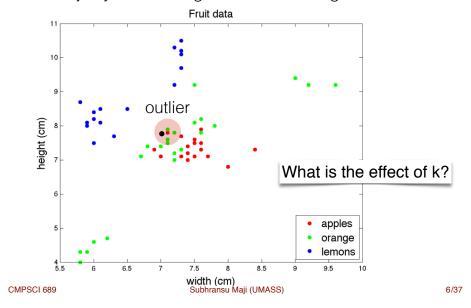


k-Nearest neighbor classifier

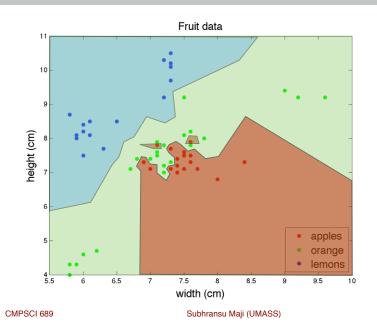


6/37

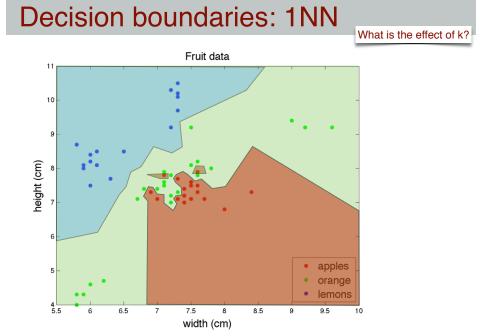
Take majority vote among the k nearest neighbors



Decision boundaries: 1NN

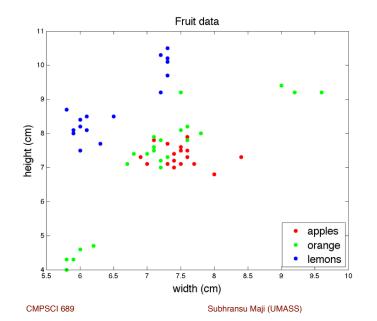


Decision boundaries: DT



Subhransu Maji (UMASS)

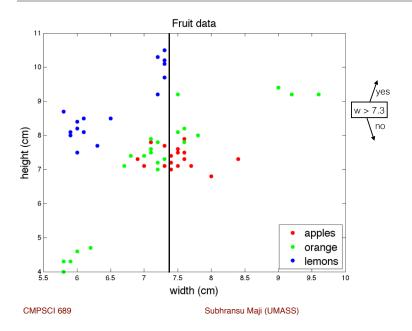
CMPSCI 689



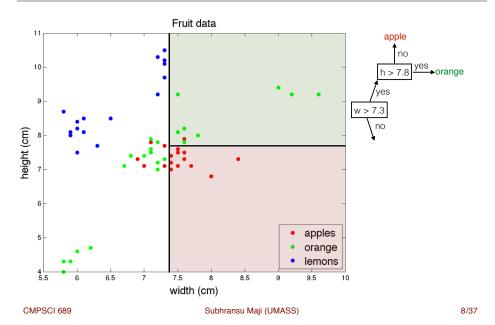
8/37

7/37

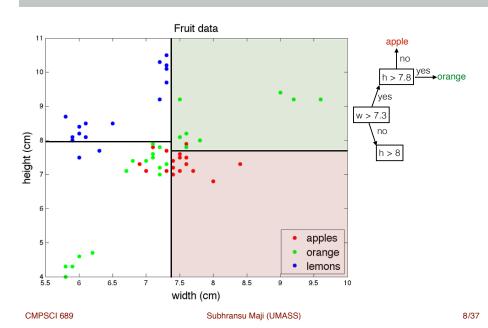
Decision boundaries: DT



Decision boundaries: DT

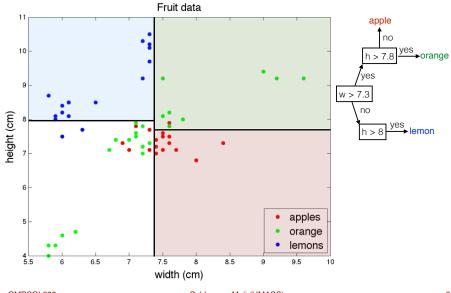


Decision boundaries: DT



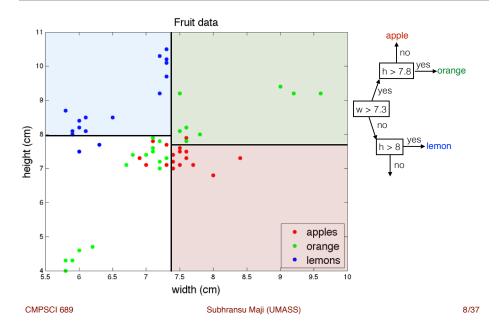
Decision boundaries: DT

8/37

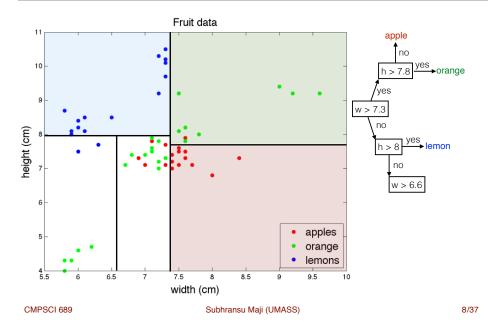


CMPSCI 689 Subhransu Maji (UMASS) 8/37

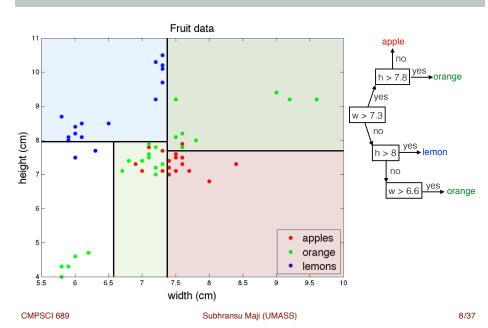
Decision boundaries: DT



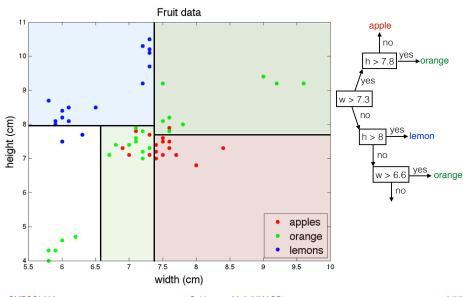
Decision boundaries: DT



Decision boundaries: DT

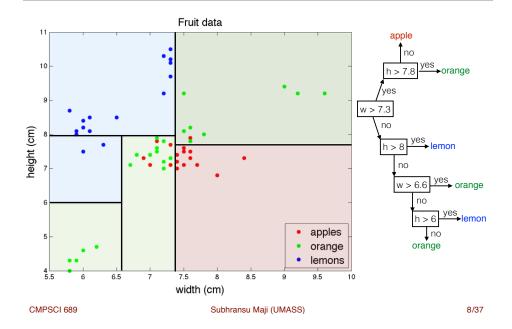


Decision boundaries: DT



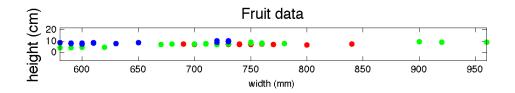
CMPSCI 689 Subhransu Maji (UMASS) 8/37

Decision boundaries: DT



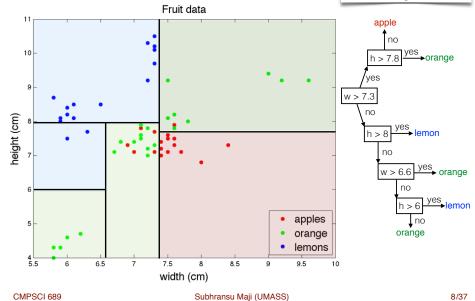
Inductive bias of the kNN classifier

- ◆ Choice of features
 - We are assuming that all features are equally important
 - ▶ What happens if we scale one of the features by a factor of 100?
- ◆ Choice of distance function
- ▶ Euclidean, cosine similarity (angle), Gaussian, etc ...
- Should the coordinates be independent?
- ◆ Choice of k



Decision boundaries: DT

The decision boundaries are axis aligned for DT



An example

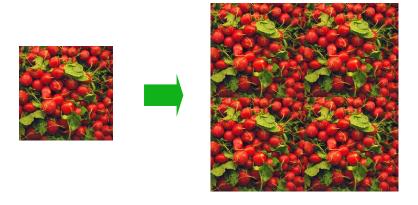
◆ "Texture synthesis" [Efros & Leung, ICCV 99]



CMPSCI 689 Subhransu Maji (UMASS) 9/37 CMPSCI 689 Subhransu Maji (UMASS) 10/37

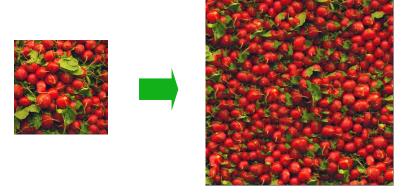
An example

◆ "Texture synthesis" [Efros & Leung, ICCV 99]



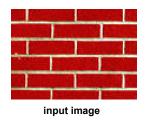
An example

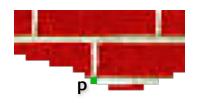
• "Texture synthesis" [Efros & Leung, ICCV 99]



 CMPSCI 689
 Subhransu Maji (UMASS)
 10/37
 CMPSCI 689
 Subhransu Maji (UMASS)
 10/37

An example: Synthesizing one pixel

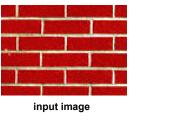


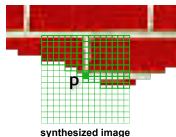


synthesized image

- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- To synthesize x
 - → pick one matching window at random
 - → assign **x** to be the center pixel of that window
- An exact match might not be present, so find the best matches using Euclidean distance and randomly choose between them, preferring better matches with higher probability

An example: Synthesizing one pixel





- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- ▶ To synthesize x
 - → pick one matching window at random
 - → assign **x** to be the center pixel of that window
- An exact match might not be present, so find the best matches using Euclidean distance and randomly choose between them, preferring better matches with higher probability

Slide from Alyosha Efros, ICCV 1999

CMPSCI 689

Subhransu Maji (UMASS)

Slide from Alyosha Efros, ICCV 1999

CMPSCI 689

Subhransu Maji (UMASS)

Slide from Alyosha Efros, ICCV 1999

Slide from Alyosha Efros, ICCV 1999

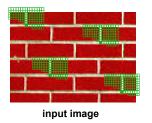
Subhransu Maji (UMASS)

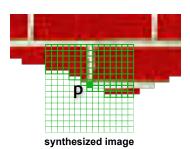
11/37

Subhransu Maji (UMASS)

11/37

An example: Synthesizing one pixel





- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- To synthesize x
 - → pick one matching window at random
 - assign x to be the center pixel of that window
- An exact match might not be present, so find the best matches using Euclidean distance and randomly choose between them, preferring better matches with higher probability

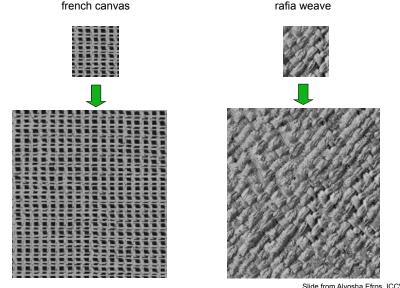
Slide from Alyosha Efros, ICCV 1999

11/37

CMPSCI 689

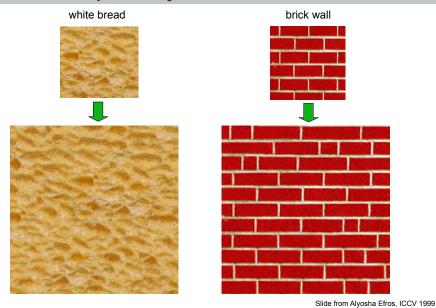
Subhransu Maji (UMASS)

An example: Synthesis results



Slide from Alyosha Efros, ICCV 1999
CMPSCI 689
Subhransu Maji (UMASS)
12/37

An example: Synthesis results



An example: Synthesis results

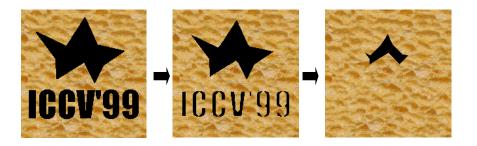
r Dick Gephardt was fai rful riff on the looming inly asked, "What's your tions?" A heartfelt sigh story about the emergenes against Clinton. "Boy g people about continuin rdt began, patiently obs, that the legal system he with this latest tancer

thaim, them . "Whephartfe lartifelintomimen el ck Clirticout omaim thartfelins faut sanesto the ry onst wartfe lck Gephttoomimeationl sigab Clicoufu Clinut Cll riff on, hat's yordn, parut tly ons ycontonsteht wasked, paim t sahe loo riff on l nskoneploourtfeas leil A nst Clit, "Wleontongal s k Cirticouirtfepe ong pme abegal fartfenstemem itiensteneltorydt telemephinsperdt was agemer ff ons artientont Cling peme as rtfe atith, "Boui s nal s fartfelt sig pedril dt ske abounutie aboutioo tfeonewas you abownthardt thatins fain, ped, ains, them, pabout wasy artiit couitly d, In A h ole emthrdngbooreme agas fa bontinsyst Clinut ory about continst Clipeopinst Cloke agatiff out (stome ginemen fly ardt beoraboul n, thenly as t C cons faimeme Diontont wat coutlyohgans as fan ien, phrtfaul, "Wbout cout congagal comininga mifmst Cliny abon al coountha.emungairt tf oun Whe looorystan loontieph. Intly on, theoplegatick (iul tatiesontly atie Diontiomt wal s f tbegåe ener mthahgat's enenhinmas fan, "intchthory ahons y

Slide from Alyosha Efros, ICCV 1999

An example: Growing Texture

- Starting from the initial image, "grow" one pixel at a time
 - Application: remove an object from the image



Slide from Alyosha Efros, ICCV 1999 Subhransu Maji (UMASS)

15/37

Subhransu Maji (UMASS)

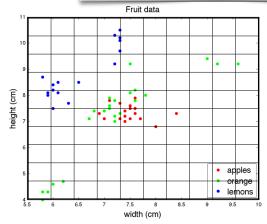
16/37

Practical issues when using kNN

- ◆ Curse of dimensionality
- ◆ Speed

CMPSCI 689

How many neighborhoods are there?



#bins = 10x10d = 2

Practical issues when using kNN

Practical issues when using kNN

◆ Curse of dimensionality

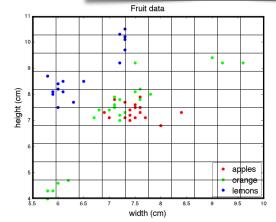
Curse of dimensionality

Speed

◆ Speed

CMPSCI 689

How many neighborhoods are there?



#bins = 10x10d = 2

 $\#bins = 10^d$ d = 1000

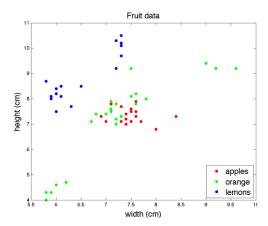
Atoms in the universe $\sim 10^{80}$

CMPSCI 689 Subhransu Maji (UMASS) 17/37 CMPSCI 689 Subhransu Maji (UMASS) 17/37

Practical issues when using kNN

- Curse of dimensionality
- Speed
 - ▶ Time taken by kNN for N points of D dimensions
 - → time to compute distances: O(ND)
 - time to find the k nearest neighbor
 - O(k N): repeated minima
 - O(N log N): sorting
 - O(N + k log N) : min heap
 - O(N + k log k) : fast median
 - → Total time is dominated by distance computation
 - We can be faster if we are willing to sacrifice exactness

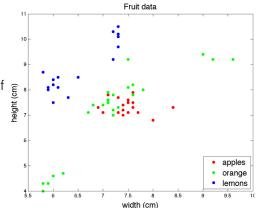
Approximate kNN



CMPSCI 689 Subhransu Maji (UMASS) 18/37

Approximate kNN

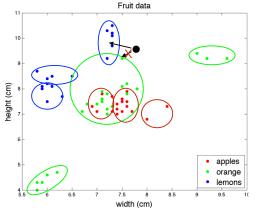
- Simplest idea is to cluster the data
 - Class → 3 clusters
 - Cluster → mean of points
 - Label of a test is the label of the nearest cluster mean



CMPSCI 689 Subhransu Maji (UMASS) 19/37

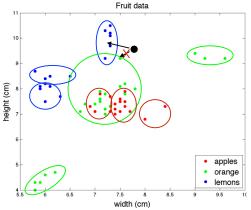
Approximate kNN

- Simplest idea is to cluster the data
 - Class → 3 clusters
 - Cluster → mean of points
 - Label of a test is the label of the nearest cluster mean



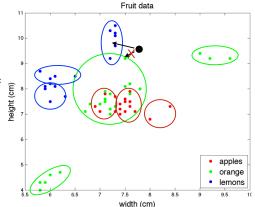
 CMPSCI 689
 Subhransu Maji (UMASS)
 19/37
 CMPSCI 689
 Subhransu Maji (UMASS)
 19/37

- Simplest idea is to cluster the data
 - Class → 3 clusters
 - Cluster → mean of points
- ▶ Label of a test is the label of the nearest cluster mean
- Run time memory
 - O(NO) O(CD)
 - → C << N



Approximate kNN

- ◆ Simplest idea is to cluster the data
- Class → 3 clusters
- Cluster → mean of points
- Label of a test is the label of the nearest cluster mean
- ◆ Run time memory
 - O(NO) O(CD)
 - → C << N



How do we cluster the data?

CMPSCI 689 Subhransu Maji (UMASS) 19/37 CMPSCI 689 Subhransu Maji (UMASS) 19/37

Clustering using k-means

Given $(x_1, x_2, ..., x_n)$, k-means clustering aims to partition the **n** observations into **k** (\leq **n**) sets **S** = { S_1 , S_2 , ..., S_k } so as to minimize the within-cluster sum of squares.

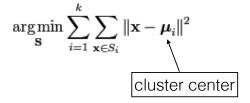
In other words, its objective is to find:

$$\underset{\mathbf{S}}{\operatorname{arg\,min}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$
 cluster center

Clustering using k-means

Given $(x_1, x_2, ..., x_n)$, k-means clustering aims to partition the **n** observations into **k** (\leq **n**) sets **S** = { S_1 , S_2 , ..., S_k } so as to minimize the within-cluster sum of squares.

In other words, its objective is to find:



Easy to compute μ given **S** and vice versa.

Lloyd's algorithm for k-means

- ◆ Initialize k centers by picking k points randomly
- ◆ Repeat till convergence (or max iterations)
 - Assign each point to the nearest center (assignment step)

$$\operatorname*{arg\,min}_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

• Estimate the mean of each group (update step)

$$\underset{\mathbf{S}}{\operatorname{arg\,min}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \underline{\boldsymbol{\mu}}_i\|^2$$

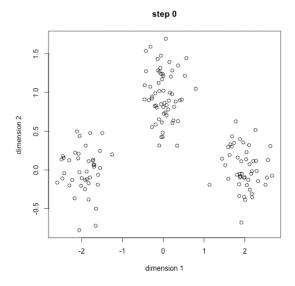
- Simple and works well in practice
 - Multiple initializations
 - Provably fast

CMPSCI 689

Subhransu Maji (UMASS)

21/37

K-means in action

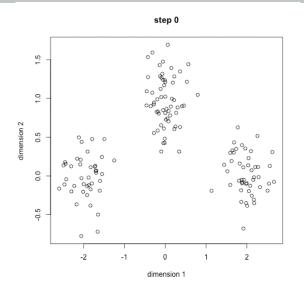


22/37

http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

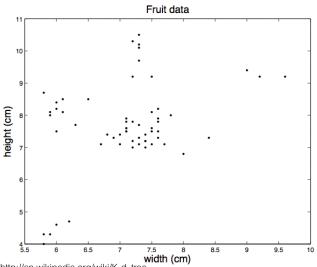
CMPSCI 689 Subhransu Maji (UMASS)

K-means in action



Approximate kNN

◆ k-d tree: O(log N) query time



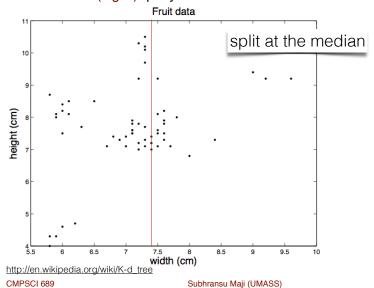
http://en.wikipedia.org/wiki/K-d_tree

Subhransu Maji (UMASS)

http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/

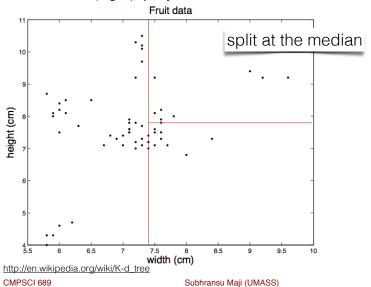
CMPSCI 689 Subhransu Maji (UMASS) 22/37 CMPSCI 689 23/37

◆ k-d tree: O(log N) query time



Approximate kNN

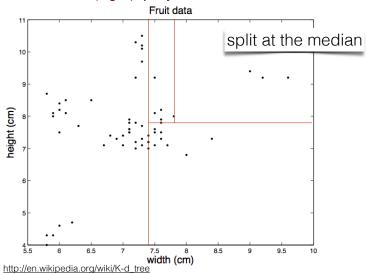
◆ k-d tree: O(log N) query time



Approximate kNN

◆ k-d tree: O(log N) query time

CMPSCI 689



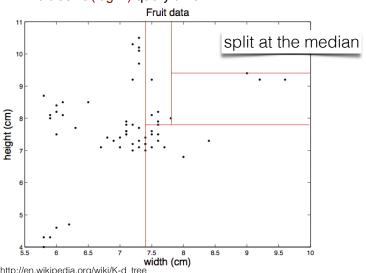
Subhransu Maji (UMASS)

Approximate kNN

23/37

23/37

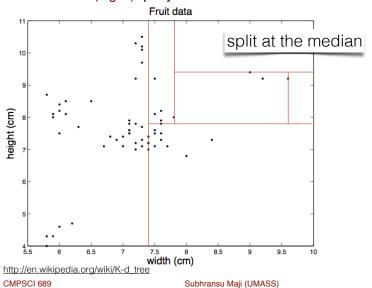
◆ k-d tree: O(log N) query time



http://en.wikipedia.org/wiki/K-d_tree

CMPSCI 689 Subhransu Maji (UMASS) 23/37

◆ k-d tree: O(log N) query time



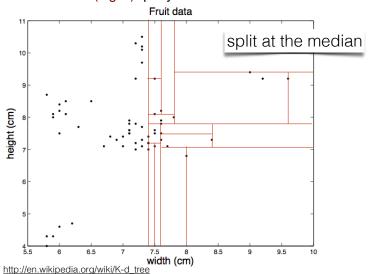
23/37

23/37

Approximate kNN

◆ k-d tree: O(log N) query time

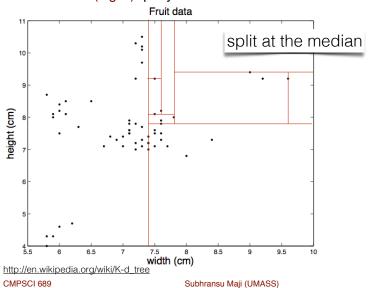
CMPSCI 689



Subhransu Maji (UMASS)

Approximate kNN

◆ k-d tree: O(log N) query time

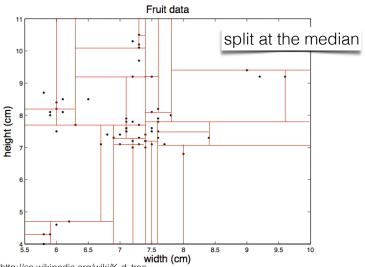


23/37

23/37

Approximate kNN

◆ k-d tree: O(log N) query time

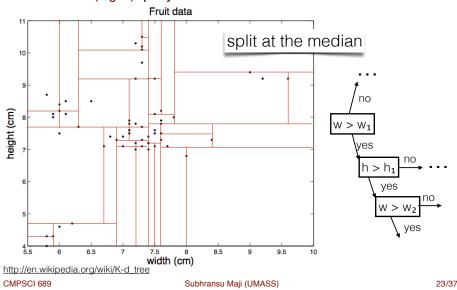


http://en.wikipedia.org/wiki/K-d_tree

CMPSCI 689 Subhransu Maji (UMASS)

Decision trees?

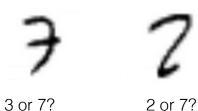
◆ k-d tree: O(log N) query time



Not everything is learnable

- It may not be possible to get perfect classification on data
 - Measurement noise: sensors may be inaccurate
 - Information gap: Sometimes we just don't have enough information to make accurate predictions
 - → e.g. Class ratings have high variance
 - Will students like AI? (70% yes, 30% no)

→ e.g. Image



The best error you can get is called the Bayes error

Summary of kNN

- Very simple setup
- Training: none
- ▶ Testing: find k nearest neighbors and take the majority class label
- ◆ An example of a non-parametric classifier: the number of parameters of the classifier *grow* with the size of the training data
- ◆ Practical issues
- ➤ Curse of dimensionality: worst case dataset size grows O(n^d)
- ▶ Speed: clustering (using k-means) and k-d trees as approximations
- ◆ kNN is likely to be competitive when:
 - ▶ the number of features are relatively small (< 20)</p>
- ▶ the distance metric is good
- ▶ the dataset is large
- Research questions:
 - Learning a good metric
 - ▶ Testing speed: RP trees, locality sensitive hashing (LSH),

CMPSCI 689 Subhransu Maji (UMASS) 24/37

Lets do a bit of learning theory ...

 CMPSCI 689
 Subhransu Maji (UMASS)
 25/37

Bayes optimal classifier and error

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

CMPSCI 689

Subhransu Maji (UMASS)

27/37

CMPSCI 689

Subhransu Maji (UMASS)

27/37

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

 $\ell(y,\hat{y})$: loss function

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

 $\ell(y,\hat{y})$: loss function

 $\epsilon(\hat{y}) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\ell(y,\hat{y})\right]$: expected error of a predictor

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

 $\ell(y,\hat{y})$: loss function

 $\epsilon(\hat{y}) = \mathbb{E}_{(\mathbf{x},y)\sim\ D}\left[\ell(y,\hat{y})\right]$: expected error of a predictor

 $\epsilon(\mathbf{x},\hat{y}) = \mathbb{E}_{y \sim \ D(y;\mathbf{x})} \left[\ell(y,\hat{y}) \right]$: expected error of a predictor at \mathbf{x}

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

 $\ell(y,\hat{y})$: loss function

 $\epsilon(\hat{y}) = \mathbb{E}_{(\mathbf{x},y) \sim \ D}\left[\ell(y,\hat{y})\right]$: expected error of a predictor

 $\epsilon(\mathbf{x},\hat{y}) = \mathbb{E}_{y \sim \ D(y;\mathbf{x})} \left[\ell(y,\hat{y}) \right]$: expected error of a predictor at \mathbf{x}

 $y^*(\mathbf{x}) = \arg\min_{\hat{y}} \epsilon(\mathbf{x}, \hat{y})$: Bayes optimal classifier

 $\epsilon^*(\mathbf{x}) = \epsilon(\mathbf{x}, y^*)$: Bayes error

CMPSCI 689

Subhransu Maji (UMASS)

27/37

CMPSCI 689

Subhransu Maji (UMASS)

27/37

Bayes optimal classifier and error

 $(\mathbf{x},y) \sim D(\mathbf{x},y)$: training data

 $\ell(y,\hat{y})$: loss function

 $\epsilon(\hat{y}) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\ell(y,\hat{y})\right]$: expected error of a predictor

 $\epsilon(\mathbf{x},\hat{y}) = \mathbb{E}_{y \sim \ D(y;\mathbf{x})} \left[\ell(y,\hat{y}) \right]$: expected error of a predictor at \mathbf{x}

 $y^*(\mathbf{x}) = \arg\min_{\hat{y}} \epsilon(\mathbf{x}, \hat{y})$: Bayes optimal classifier

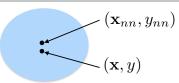
 $\epsilon^*(\mathbf{x}) = \epsilon(\mathbf{x}, y^*)$: Bayes error

Binary classification $y \in \{0,1\}$ $\ell(y,\hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{otherwise} \end{cases}$

$$y^*(\mathbf{x}) = \arg\min_{\hat{y}} [D(y=0; \mathbf{x})\ell(0, \hat{y}) + D(y=1; \mathbf{x})\ell(1, \hat{y})]$$

$$y^*(\mathbf{x}) = \begin{cases} 0 & \text{if } D(y = 0; \mathbf{x}) \ge 0.5 & \epsilon^*(\mathbf{x}) = 1 - D(y^*(\mathbf{x}); \mathbf{x}) \\ 1 & \text{if } D(y = 0; \mathbf{x}) < 0.5 \end{cases}$$

NN classifier is nearly optimal

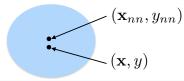


$$As n \to \infty$$

$$D(y_{nn}; \mathbf{x}_{nn}) \to D(y; \mathbf{x})$$

$$\begin{aligned}
\epsilon_{nn}^{1}(\mathbf{x}) &= P(y=1, y_{nn}=0; \mathbf{x}, \mathbf{x}_{nn}) + P(y=0, y_{nn}=1; \mathbf{x}, \mathbf{x}_{nn}) \\
&= D(y=1; \mathbf{x}) D(y_{nn}=0; \mathbf{x}_{nn}) + D(y=0; \mathbf{x}) D(y_{nn}=1; \mathbf{x}_{nn}) \\
&= 2D(y=1; \mathbf{x}) D(y=0; \mathbf{x}) \\
&\leq 2 \min \left(D(y=1; \mathbf{x}), D(y=0; \mathbf{x}) \right) \\
&= 2\epsilon^{*}(\mathbf{x})
\end{aligned}$$

NN classifier is nearly optimal



As
$$n \to \infty$$

 $D(y_{nn}; \mathbf{x}_{nn}) \to D(y; \mathbf{x})$

$$\epsilon_{nn}^{1}(\mathbf{x}) = P(y = 1, y_{nn} = 0; \mathbf{x}, \mathbf{x}_{nn}) + P(y = 0, y_{nn} = 1; \mathbf{x}, \mathbf{x}_{nn})$$

$$= D(y = 1; \mathbf{x})D(y_{nn} = 0; \mathbf{x}_{nn}) + D(y = 0; \mathbf{x})D(y_{nn} = 1; \mathbf{x}_{nn})$$

$$= 2D(y = 1; \mathbf{x})D(y = 0; \mathbf{x})$$

$$\leq 2\min(D(y = 1; \mathbf{x}), D(y = 0; \mathbf{x}))$$

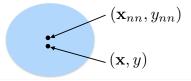
$$= 2\epsilon^{*}(\mathbf{x})$$

$$\epsilon^* \leq \epsilon_{nn}^1 \leq 2\epsilon^*$$
 Cover-Hart, 1967

CMPSCI 689 Subhransu Maji (UMASS) 28/37

Machine learning solved?

NN classifier is nearly optimal



As
$$n \to \infty$$

 $D(y_{nn}; \mathbf{x}_{nn}) \to D(y; \mathbf{x})$

$$\begin{aligned}
\epsilon_{nn}^{1}(\mathbf{x}) &= P(y=1, y_{nn}=0; \mathbf{x}, \mathbf{x}_{nn}) + P(y=0, y_{nn}=1; \mathbf{x}, \mathbf{x}_{nn}) \\
&= D(y=1; \mathbf{x}) D(y_{nn}=0; \mathbf{x}_{nn}) + D(y=0; \mathbf{x}) D(y_{nn}=1; \mathbf{x}_{nn}) \\
&= 2D(y=1; \mathbf{x}) D(y=0; \mathbf{x}) \\
&\leq 2 \min \left(D(y=1; \mathbf{x}), D(y=0; \mathbf{x}) \right) \\
&= 2\epsilon^{*}(\mathbf{x})
\end{aligned}$$

$$\epsilon^* \le \epsilon_{nn}^1 \le 2\epsilon^*$$
 Cover-Hart, 1967

For any
$$k \ge 5$$
, $\epsilon^* \le \epsilon_{nn}^k \le \epsilon^* \left(1 + \sqrt{\frac{2}{k}}\right)$

Subhransu Maji (UMASS)

28/37

Not really ...

- ♦ kNN is nearly optimal when there is infinite training data
 - Says nothing about the finite sample case
 - Note: not all classifiers are (nearly) optimal even with infinite data
- ◆ Bayes error is a function of features (x)
 - We can get better **Bayes error** if we choose different the features
 - → If we had color in addition to the width and height, we would be able classify the fruits more accurately.
- ◆ How do we understand the performance of learners for the finite sample case?
 - Bias-variance decomposition

29/36 CMPSCI 689 Subhransu Maji (UMASS) 30/37

Bias-variance decomposition

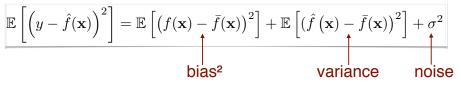
• Standard way to decompose squared loss $\ell(y,\hat{y}) = (y-\hat{y})^2$

$$y = f(\mathbf{x}) + \epsilon$$
 $\epsilon \sim N(0; \sigma^2)$ true function noise

$$\left| (\mathbf{x}_1,y_1), (\mathbf{x}_2,y_2), \dots, (\mathbf{x}_n,y_n)
ightarrow \hat{f}(\mathbf{x})
ight| \; \; ext{training algorithm}$$

 $\left|ar{f}(\mathbf{x}) = \mathbb{E}\hat{f}(\mathbf{x})
ight|$ expectation of the learned function

expectation is over datasets



CMPSCI 689 Subhransu Maji (UMASS)

 $\epsilon = N(0, \sigma^2)$ $\sigma = 0.1$

 $y = f(x) + \epsilon$ $f(x) = \sin(\pi x)$

Example: curve fitting

figures from https://theclevermachine.wordpress.com/tag/estimator-variance/ CMPSCI 689 Subhransu Maji (UMASS)

Example: curve fitting

$$y = f(x) + \epsilon \qquad f(x) = \sin(\pi x)$$

$$\epsilon = N(0, \sigma^2) \quad \sigma = 0.1$$

$$0.1$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

$$0.5$$

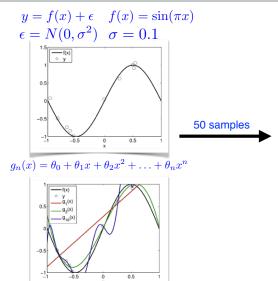
$$0.5$$

$$0.5$$

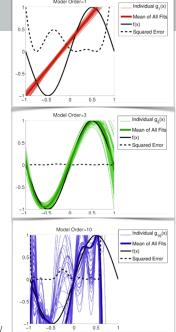
$$0$$

figures from https://theclevermachine.wordpress.com/tag/estimator-variance/ Subhransu Maji (UMASS)

Example: curve fitting



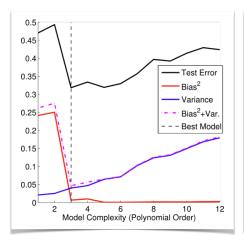
figures from https://theclevermachine.wordpress.com/tag/estimator-variance/ CMPSCI 689 Subhransu Maji (UMASS)

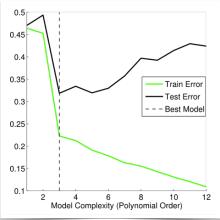


32/37

31/37

Example: curve fitting





Bias-variance decomposition proof

$$\mathbb{E}\left[\left(y-\hat{f}\right)\right)^{2}\right] = \mathbb{E}\left[\left(f+\epsilon-\hat{f}\right)^{2}\right]$$

$$= \mathbb{E}\left[\left(f-\hat{f}\right)^{2}\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}+\bar{f}-\hat{f}\right)^{2}\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\mathbb{E}\left[\left(f-\bar{f}\right)\left(\bar{f}-\hat{f}\right)\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\mathbb{E}\left[\left(f\bar{f}-f\hat{f}-f\hat{f}+\bar{f}\hat{f}\right)\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\left(f\bar{f}-f\bar{f}-\bar{f}\bar{f}+\bar{f}\bar{f}\right) + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + \sigma^{2}$$

figures from https://theclevermachine.wordpress.com/tag/estimator-variance/CMPSCI 689 Subhransu Maji (UMASS)

CMPSCI 689

33/37

Subhransu Maji (UMASS)

34/37

Bias-variance decomposition proof

$$\mathbb{E}\left[\left(y-\hat{f}\right)\right)^{2}\right] = \mathbb{E}\left[\left(f+\epsilon-\hat{f}\right)^{2}\right]$$

$$= \mathbb{E}\left[\left(f-\hat{f}\right)^{2}\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}+\bar{f}-\hat{f}\right)^{2}\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\mathbb{E}\left[\left(f-\bar{f}\right)\left(\bar{f}-\hat{f}\right)\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\mathbb{E}\left[\left(f\bar{f}-f\hat{f}-f\hat{f}+\bar{f}\hat{f}\right)\right] + \sigma^{2}$$

$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\left(f\bar{f}-f\bar{f}-\bar{f}\bar{f}+\bar{f}\bar{f}\right) + \sigma^{2}$$

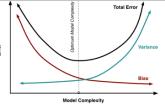
$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + 2\left(f\bar{f}-f\bar{f}-\bar{f}\bar{f}+\bar{f}\bar{f}\right) + \sigma^{2}$$

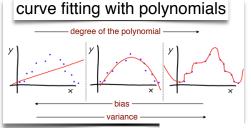
$$= \mathbb{E}\left[\left(f-\bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\bar{f}-\hat{f}\right)^{2}\right] + \sigma^{2}$$

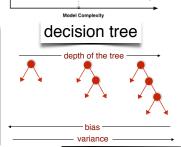
Similar decomposition can be obtained for the 0/1 loss

Bias-variance tradeoff for learners

$$\mathbb{E}\left[\left(y - \hat{f}\right)^{2}\right] = \mathbb{E}\left[\left(f - \bar{f}\right)^{2}\right] + \mathbb{E}\left[\left(\hat{f} - \bar{f}\right)^{2}\right] + \sigma^{2}$$
error = bias + variance + noise







kNN regression

k bias variance
Subhransu Maji (UMASS)

Summary

♦ kNN classifiers

- geometry, metric, decision boundaries
- effect of k
- practical issues
- curse of dimensionality
- ⇒ speed: clustering using k-means, k-d trees

◆ Theory

- Bayes optimality
- ▶ kNN is nearly Bayes optimal as training dataset size goes to infinity
- ▶ Bias-variance decomposition
- Understanding overfitting and underfitting

Slides credit

- ◆ The fruit classification dataset is from Iain Murray at University of Edinburgh — http://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/.
- ◆ The slides on texture synthesis are from Efros and Leung's ICCV 2009 presentation.
- ◆ Figures of the bias-variance tradeoff are from https://theclevermachine.wordpress.com/tag/estimator-variance/.

 CMPSCI 689
 Subhransu Maji (UMASS)
 36/37
 CMPSCI 689
 Subhransu Maji (UMASS)
 37/37