به نام یگانه معبود بخشنده مهربان

مبانی یادگیری ماشین

Machine Learning Foundations

گروه هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

ترم اول سال تحصیلی ۰۲ - ۰۳

ارائه دهنده : پیمان ادیبی

دسته‌بند نزدیکترین همسایه‌ها

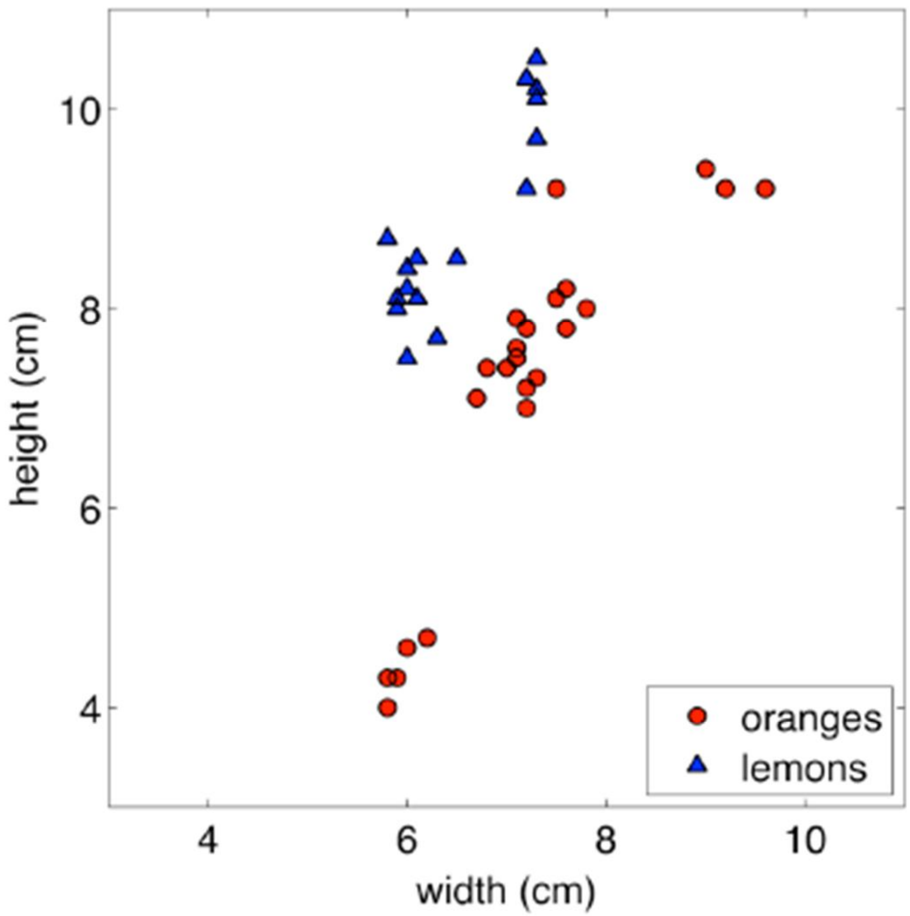
Nearest Neighbors Classifier

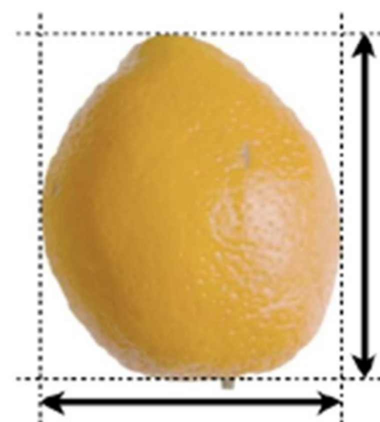
مدل‌های غیر پارامتری

- Non-parametric models
  - distance
  - non-linear decision boundaries

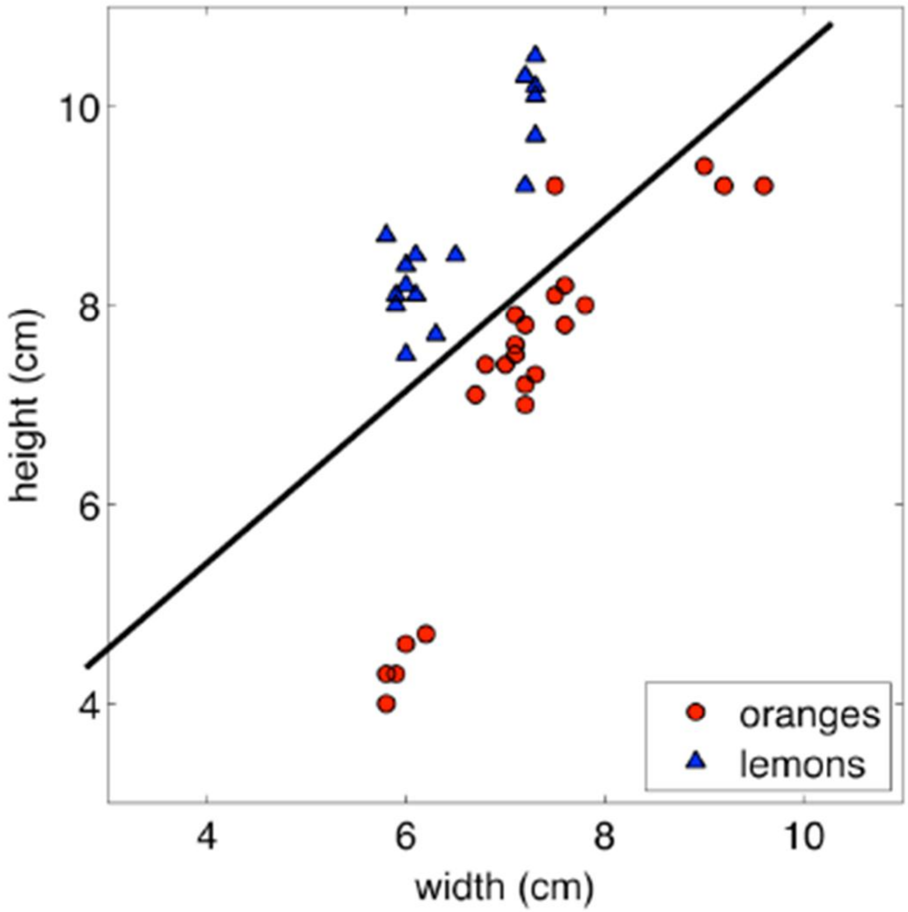
Note: We will mainly use today's method for classification, but it can also be used for regression

مثال: دسته بندی پرتقال‌ها و لیموها



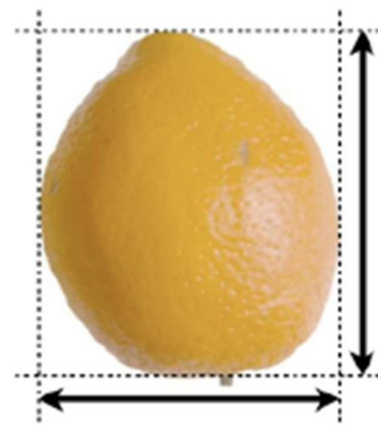


مثال: دسته بندی پرتقال‌ها و لیموها



Can construct simple linear decision boundary:

$y = \text{sign}(w_0 + w_1x_1 + w_2x_2)$



معنای دسته بند «خطی»

- Classification is intrinsically non-linear
  - It puts non-identical things in the same class, so a difference in the input vector sometimes causes zero change in the answer
- Linear classification means that the part that adapts is linear (just like linear regression)

$$z(x) = \mathbf{w}^T\mathbf{x} + w_0$$

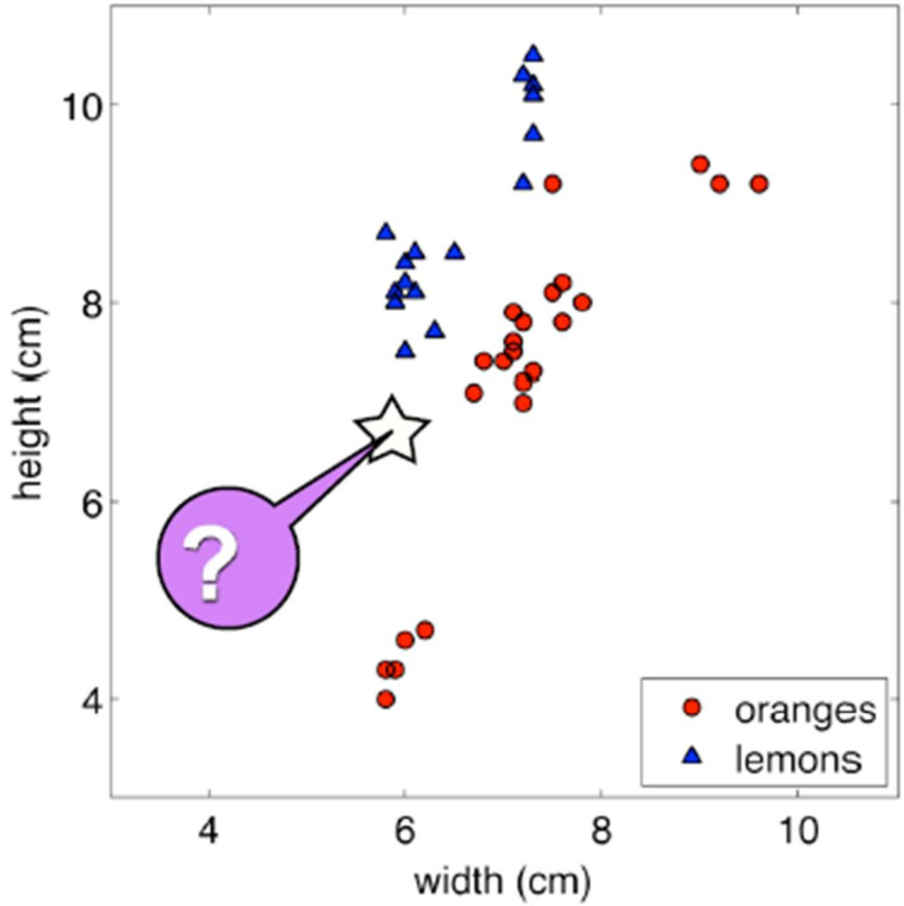
  with adaptive $\mathbf{w}, w_0$

- The adaptive part is followed by a non-linearity to make the decision

$$y(\mathbf{x}) = f(z(\mathbf{x}))$$

- What functions $f()$ have we seen so far in class?

دسته بندی به عنوان استقراء (Induction)



دسته بندی مبتنی بر نمونه (Instance-based)

- Alternative to parametric models are non-parametric models
- These are typically simple methods for approximating discrete-valued or real-valued target functions (they work for classification or regression problems)
- Learning amounts to simply storing training data
- Test instances classified using similar training instances
- Embodies often sensible underlying assumptions:
  - Output varies smoothly with input
  - Data occupies sub-space of high-dimensional input space

نزدیکترین همسایه ها

- Assume training examples correspond to points in d-dim Euclidean space
- **Idea**: The value of the target function for a new query is estimated from the known value(s) of the nearest training example(s)
- Distance typically defined to be Euclidean:

$$||\mathbf{x}^{(a)} - \mathbf{x}^{(b)}||_2 = \sqrt{\sum_{j=1}^{d}(x_j^{(a)} - x_j^{(b)})^2}$$

Algorithm:

1. Find example $(\mathbf{x}^*, t^*)$ (from the stored training set) closest to the test instance $\mathbf{x}$. That is:

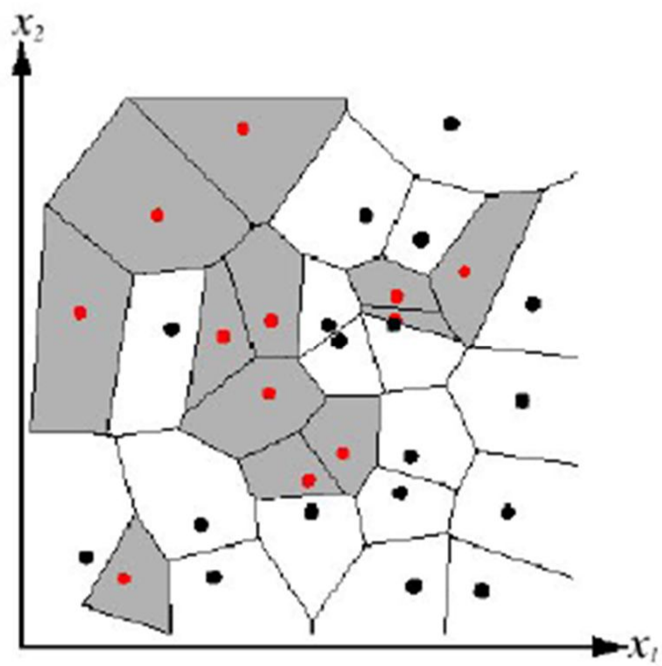
$$\mathbf{x}^* = \underset{\mathbf{x}^{(i)} \in \text{train. set}}{\operatorname{argmin}} \ \text{distance}(\mathbf{x}^{(i)}, \mathbf{x})$$

2. Output $y = t^*$

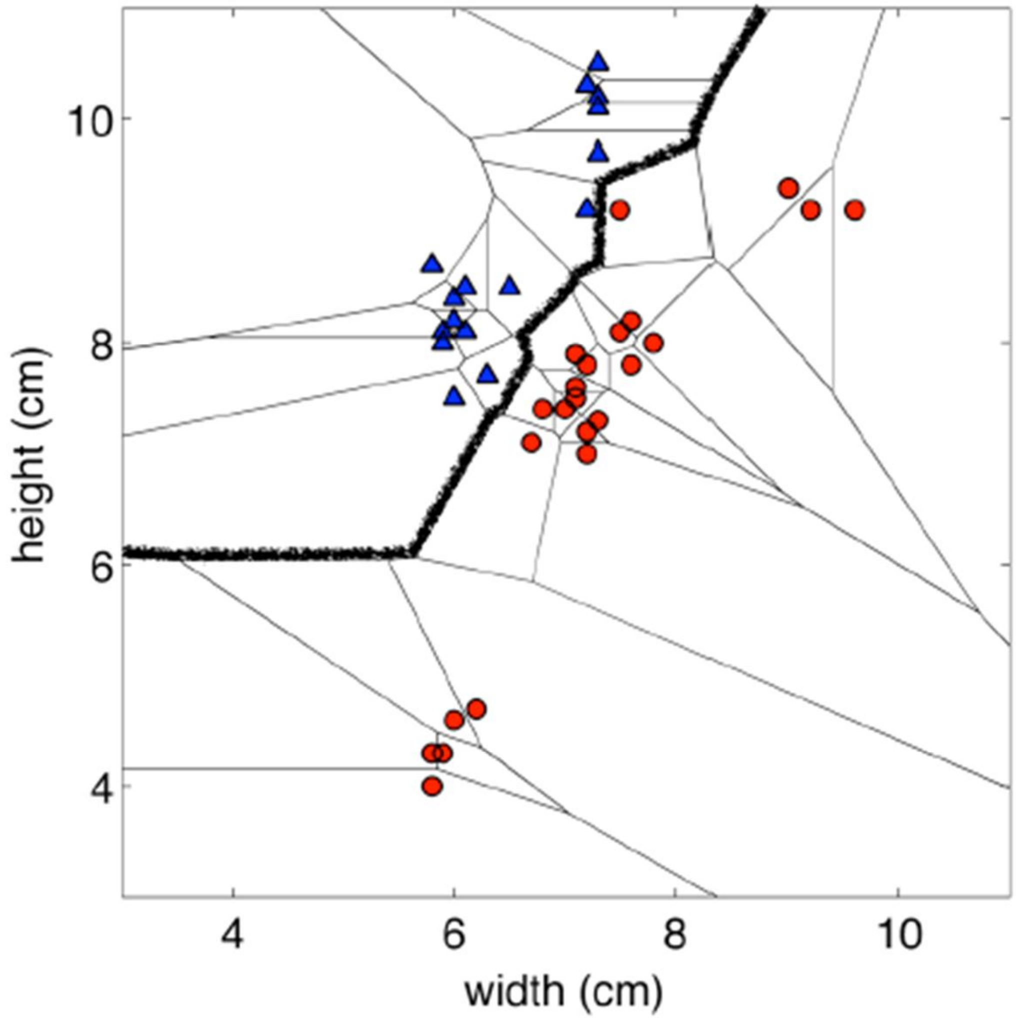
- Note: we don't really need to compute the square root. Why?

نزدیکترین همسایه ها - مرزهای تصمیم

- Nearest neighbor algorithm does not explicitly compute decision boundaries, but these can be inferred
- Decision boundaries: Voronoi diagram visualization
  - show how input space divided into classes
  - each line segment is equidistant between two points of opposite classes

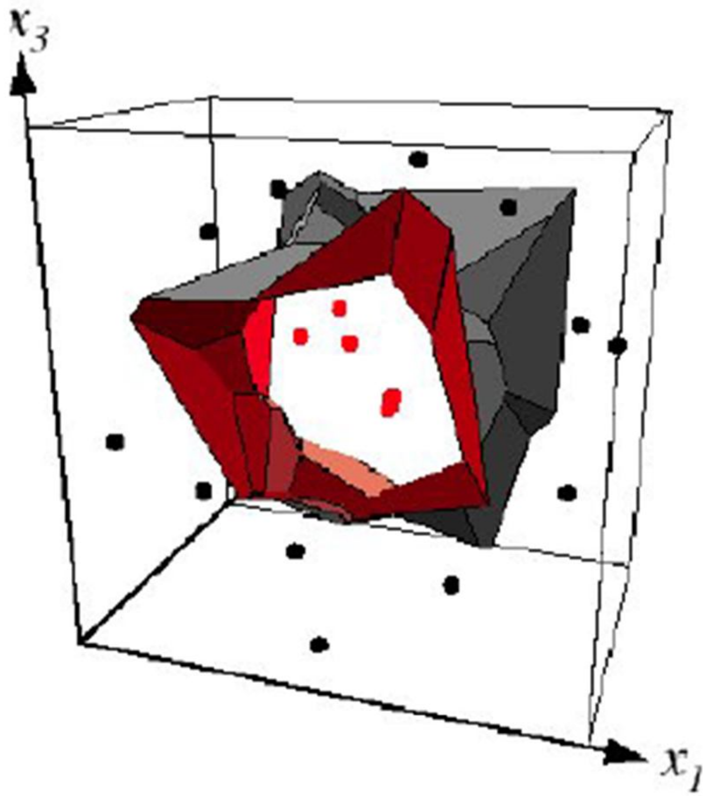


نزدیکترین همسایه ها - مرزهای تصمیم



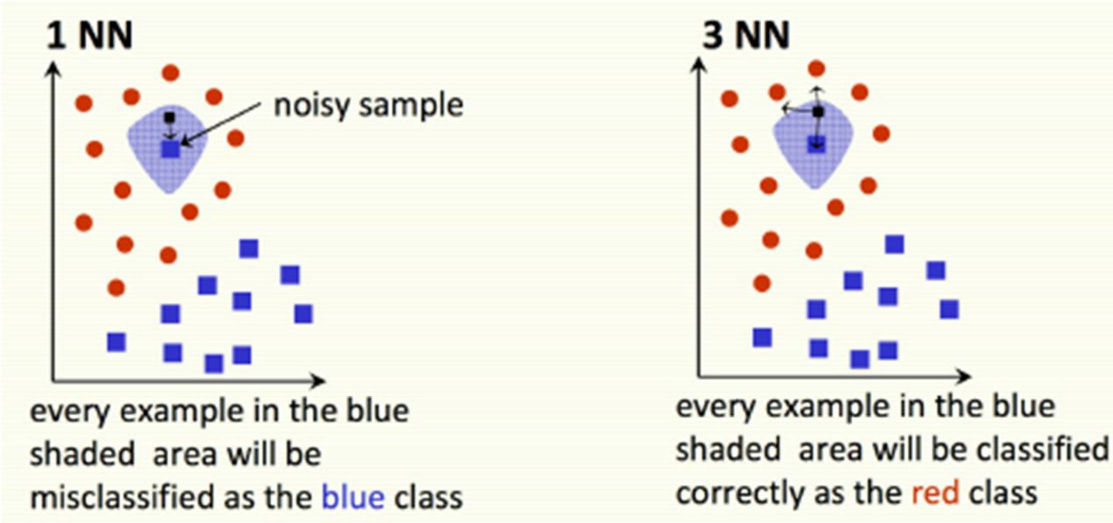
Example: 2D decision boundary

نزدیکترین همسایه ها - مرزهای تصمیم



Example: 3D decision boundary

k نزدیکترین همسایه ها (kNN)



[Pic by Olga Veksler]

- Nearest neighbors sensitive to mis-labeled data ("class noise"). Solution?
- Smooth by having k nearest neighbors vote

Algorithm (kNN):

1. Find k examples $\{\mathbf{x}^{(i)}, t^{(i)}\}$ closest to the test instance $\mathbf{x}$
2. Classification output is majority class

$$y = arg\max_{t^{(z)}} \sum_{r=1}^{k} \delta(t^{(z)}, t^{(r)})$$

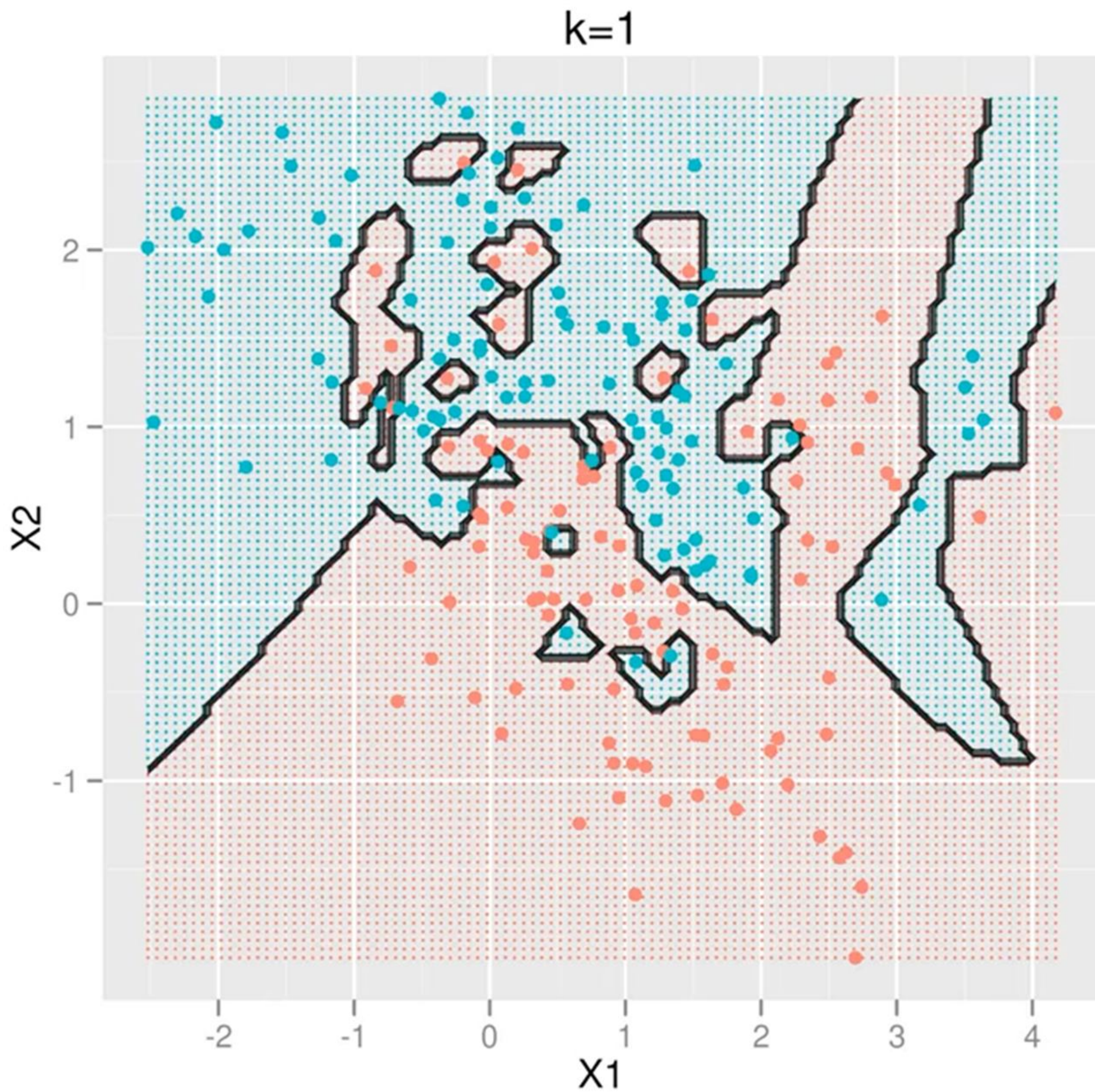
نزدیکترین همسایه ها k (kNN)

How do we choose k?

- Larger k may lead to better performance
- But if we set k too large we may end up looking at samples that are not neighbors (are far away from the query)
- We can use cross-validation to find k
- Rule of thumb is $k < sqrt(n)$, where n is the number of training examples

[Slide credit: O. Veksler]

k نزدیکترین همسایه ها (*k*NN)

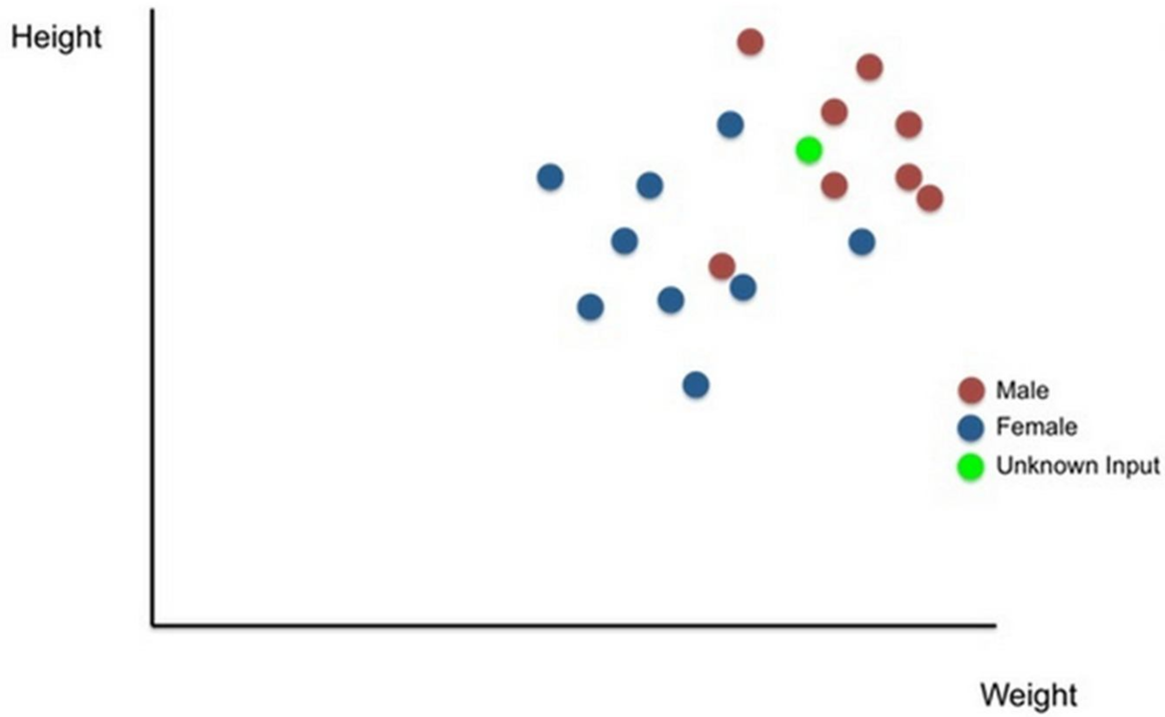


k نزدیکترین همسایه ها - مسایل و راه حل‌ها

- Some attributes have larger ranges, so are treated as more important
  - normalize scale
    - Simple option: Linearly scale the range of each feature to be, eg, in range [0,1]
    - Linearly scale each dimension to have 0 mean and variance 1 (compute mean μ and variance σ^2 for an attribute x_j and scale: $(x_j - m)/\sigma$)
  - be careful: sometimes scale matters
- Irrelevant, correlated attributes add noise to distance measure
  - eliminate some attributes
  - or vary and possibly adapt weight of attributes
- Non-metric attributes (symbols)
  - Hamming distance

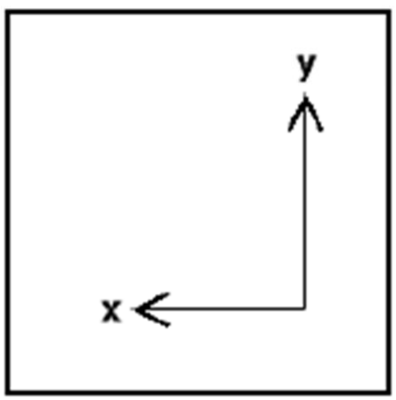
k نزدیکترین همسایه ها - مسایل و راه حل‌ها

- Assign the majority class among the k-nearest neighbors
- K=2 ?
- K=17 ?
- How do we define x^{train} nearest to x^{test}?
  - $d\left(x^{train}, x^{test}\right) \leq d(x, x^{test}), \forall x \in D$
- Obviously… but how do we define the distance function d?

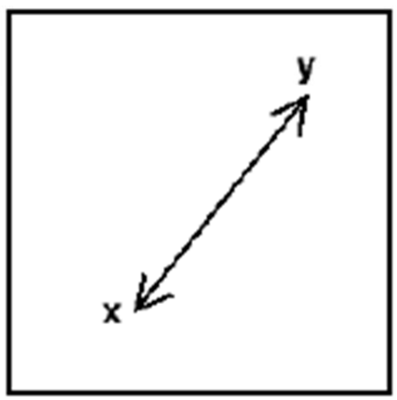


k نزدیکترین همسایه ها - مسأله فاصله‌ها

- Minkowski distance
  - $d(x, x') = \left(\sum_i |x_i - x_i'|^p\right)^{\frac{1}{p}}$
  - p is a model (K-NN) parameter
- $p = 1 \rightarrow$ (Manhattan distance)… why Manhattan?
- $p = 2 \rightarrow$(Euclidian distance)
- $p = \infty \rightarrow \max_i |x_i - x_i'|$
- $p = -\infty \rightarrow \min_i |x_i - x_i'|$



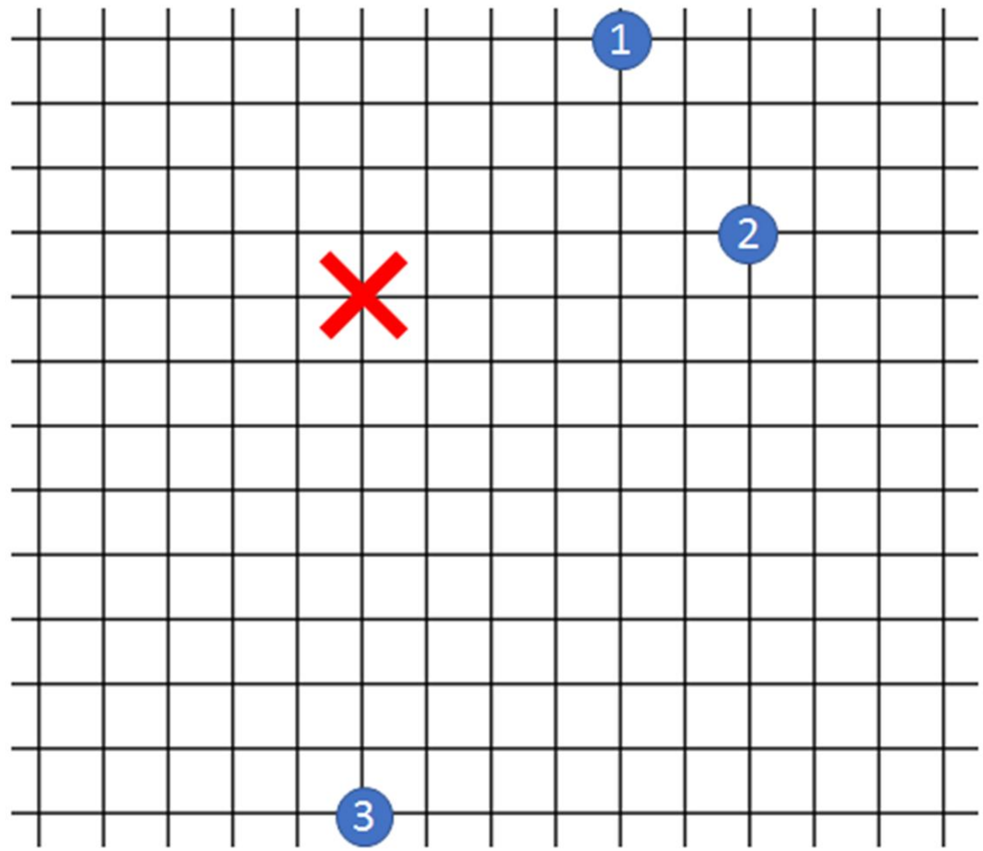
Manhattan



Euclidean

k نزدیکترین همسایه ها - مسأله فاصله‌ها

- Which blue circle is closest to the red x?
- $d(x, x') = \left(\sum_i |x_i - x'_i|^p\right)^{\frac{1}{p}}$
- $p = 1$
- $p = 2$
- $p = \infty$
- $p = -\infty$



k نزدیکترین همسایه ها - مسایل و راه حل‌ها

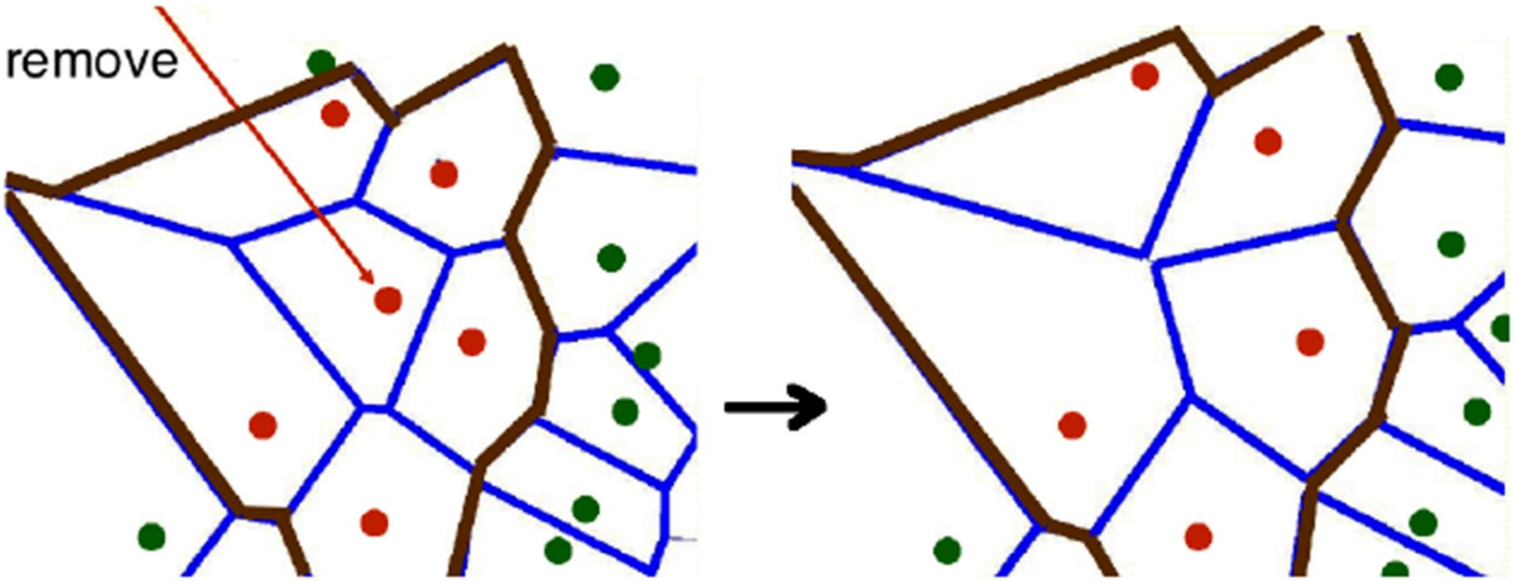
- **Expensive at test time:** To find one nearest neighbor of a query point **x**, we must compute the distance to all N training examples. Complexity: $O(kdN)$ for kNN
  - Use subset of dimensions
  - Pre-sort training examples into fast data structures (kd-trees)
  - Compute only an approximate distance (LSH)
  - Remove redundant data (condensing)
- **Storage Requirements:** Must store all training data
  - Remove redundant data (condensing)
  - Pre-sorting often increases the storage requirements
- **High Dimensional Data:** "Curse of Dimensionality"
  - Required amount of training data increases exponentially with dimension
  - Computational cost also increases dramatically

[Slide credit: David Claus]

✓380

k نزدیکترین همسایه ها – حذف افزونگی

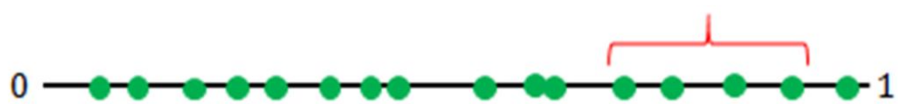
- If all Voronoi neighbors have the same class, a sample is useless, remove it



[Slide credit: O. Veksler]

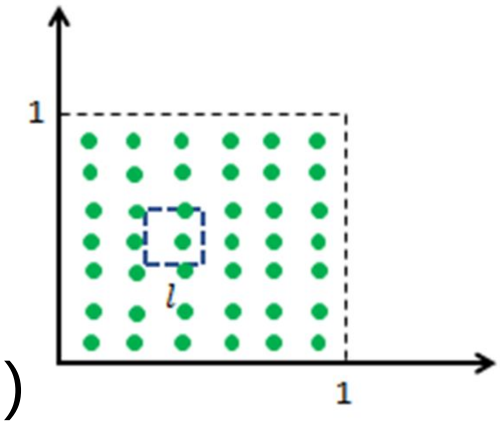
k نزدیکترین همسایه ها – تعداد داده لازم

- Assume a 1-dimension feature vector from [0,1]
- Accurate predictions require k neighbors within l distance
- Add n samples (uniform distribution)
- Interval of size l covers $\frac{l}{1}$ of the state space and should include $\frac{k}{n}$ of the (uniform) samples on expectancy
- $\frac{l}{1} = \frac{k}{n}$ and so $n = \frac{k}{l}$ e.g., $k = 10, l = 0.01$ requires 1000 samples

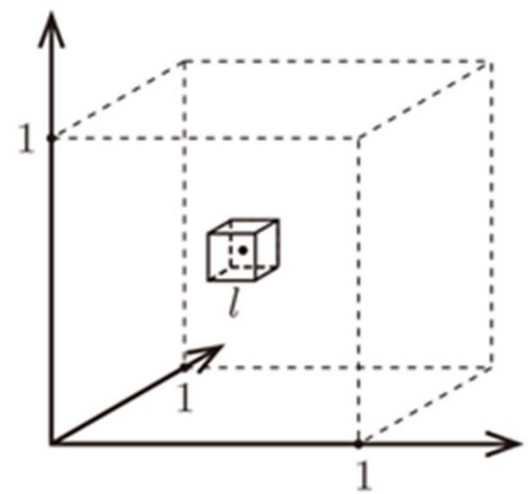


✓462

k نزدیکترین همسایه ها – تعداد داده لازم



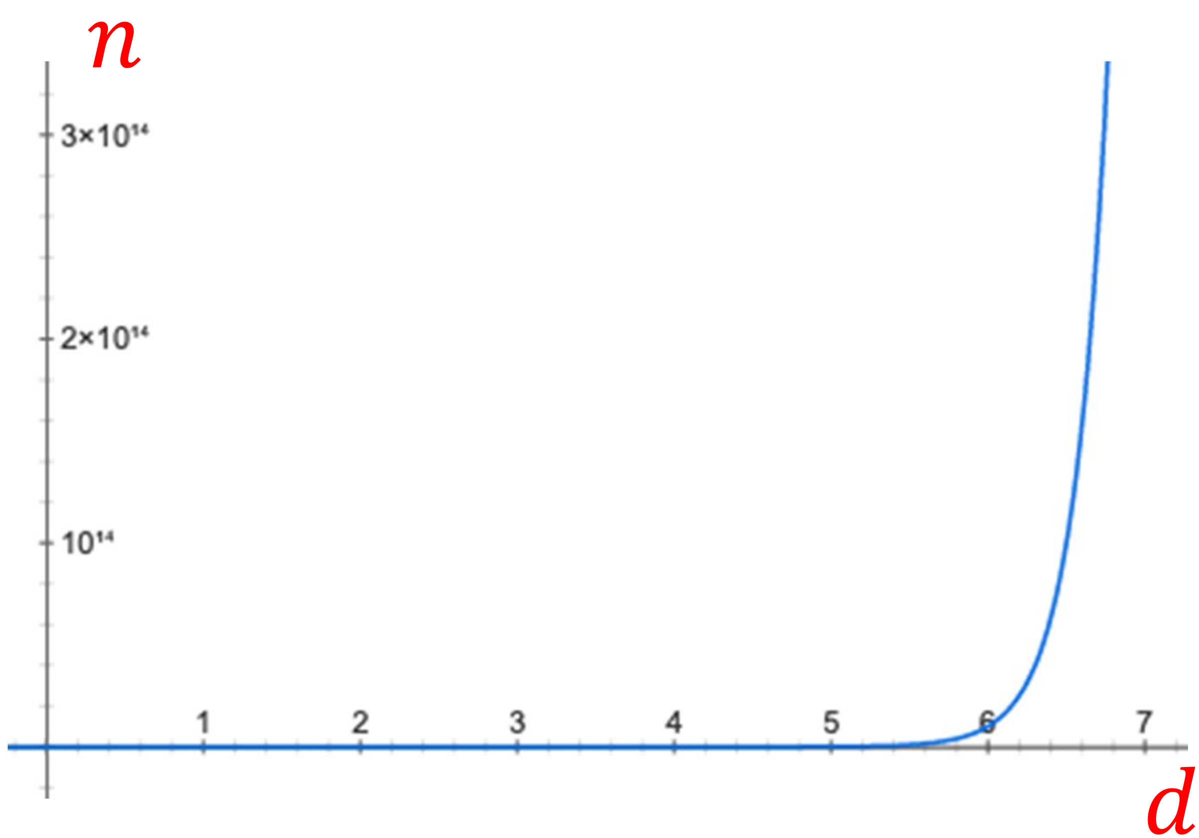
- Now consider a 2D feature vector: $\mathbb{R}^2$
- A square of size l^2 covers $\frac{l^2}{1^2}$ of the state space and should include $\frac{k}{n}$ of the (uniform) samples on expectancy → $n = \frac{k}{l^2}$
- Now consider a 3D feature vector: $[0,1]^3$ → $n = \frac{k}{l^3}$



- For the general case (d dimensions)
  - $n = \frac{k}{l^d}$

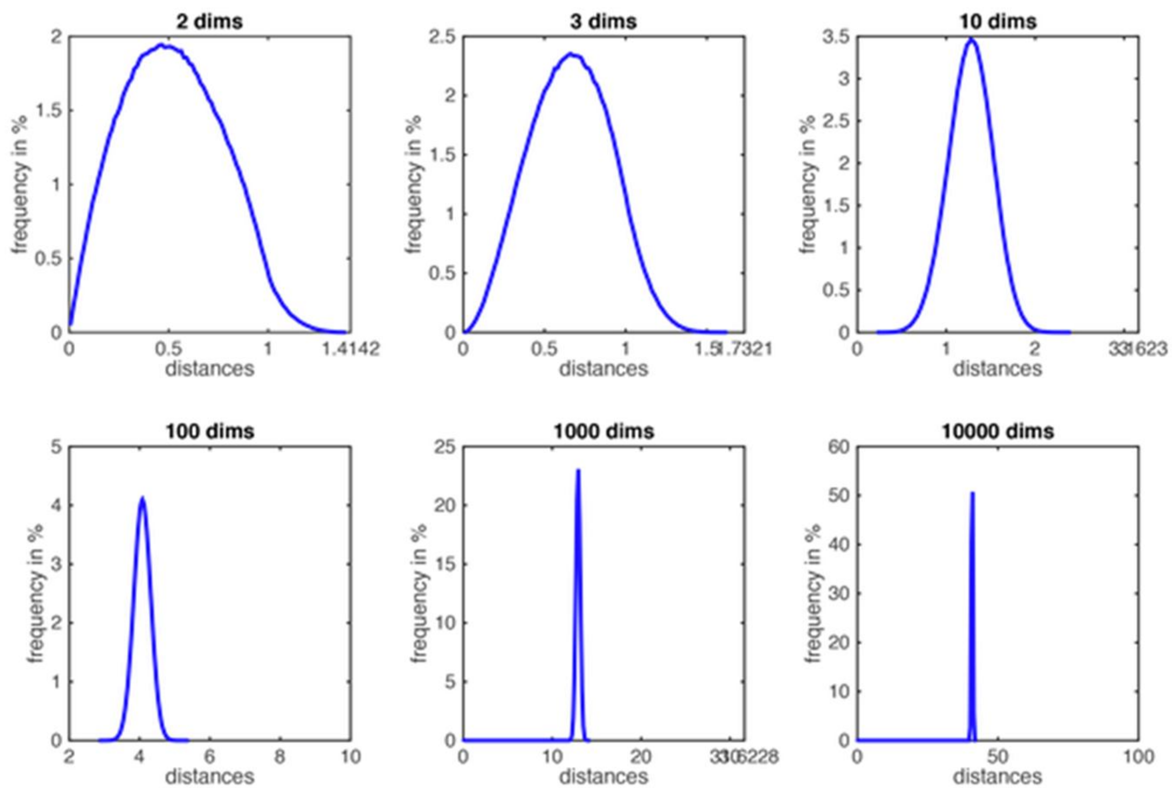
معضل بعد (Cures of Dimensionality)

- $n = \frac{k}{l^d}$
- $k = 10$
- $l = 0.1$



(Cures of Dimensionality) معضل بعد

- Frequency of pairwise Euclidian distances between randomly distributed points within d-dimensional unit squares



- K-NN is meaningless for $d \geq 10$

✓524

(Cures of Dimensionality) معضل بعد

- How many features are in a 1.3MP RGB image
  - ~4M dimensions
- We would need $\frac{10}{0.1^{4000000}}$ samples
  - Not enough atoms in the universe
- Say we could get all these samples, what is the computation complexity for inference?
  - $O(nd)$

✓526

مثال: شناسایی ارقام دستنویس

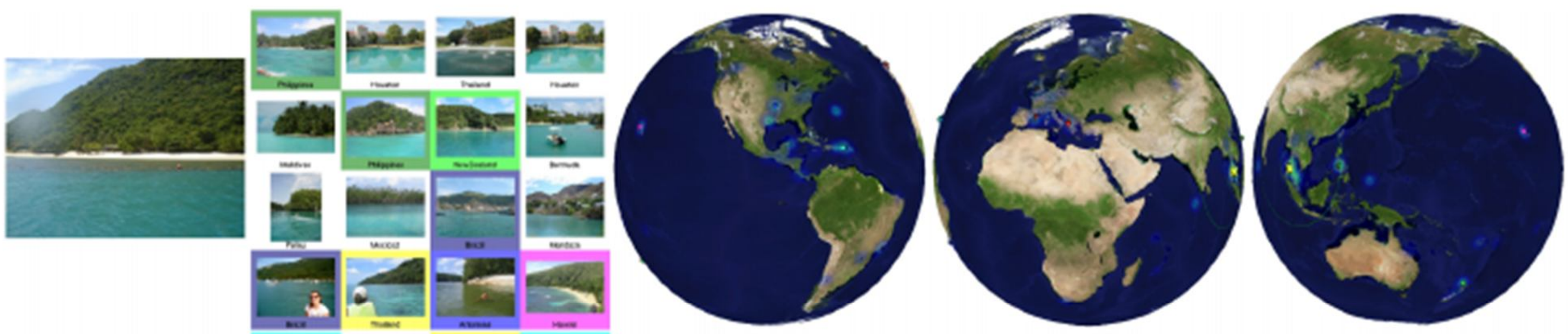
- Decent performance when lots of data

- Yann LeCunn – MNIST Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: $d = 784$
  - 60,000 training samples
  - 10,000 test samples
- Nearest neighbour is competitive

	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

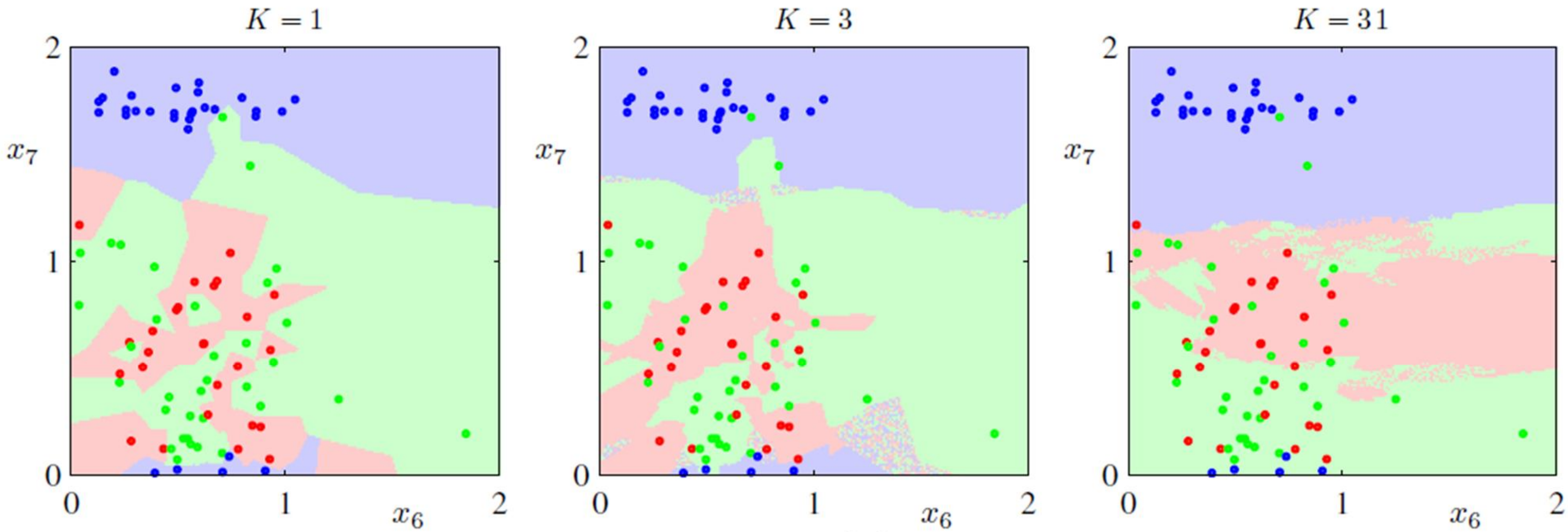
مثال: این تصویر کجای کره زمین گرفته شده؟

- Problem: Where (eg, which country or GPS location) was this picture taken?
  - Get 6M images from Flickr with gps info (dense sampling across world)
  - Represent each image with meaningful features
  - Do kNN (large k better, they use $k = 120$)!



[Paper: James Hays, Alexei A. Efros. im2gps: estimating geographic information from a single
✓526 image. CVPR'08. Project page: http://graphics.cs.cmu.edu/projects/im2gps/]

مرور



- Naturally forms complex decision boundaries; adapts to data density
- If we have lots of samples, kNN typically works well
- Problems:
  - Sensitive to class noise.
  - Sensitive to scales of attributes.
  - Distances are less meaningful in high dimensions
  - Scales linearly with number of examples

✓5

- Inductive Bias: What kind of decision boundaries do we expect to find?