Non-parametric methods
Nearest Neighbours

Natasha Jaques

UNIVERSITY of WASHINGTON

Parametric vs non-parametric

- A model is parametric if # parameters does not depend on # samples

  # Learning a bunch of parameters / weights. Like linear regression, neural networks

- A model is non-parametric if # parameters increases with # samples
  - Does not mean absence of parameters!

  # Today's class

This lecture: k nearest neighbors

Simple

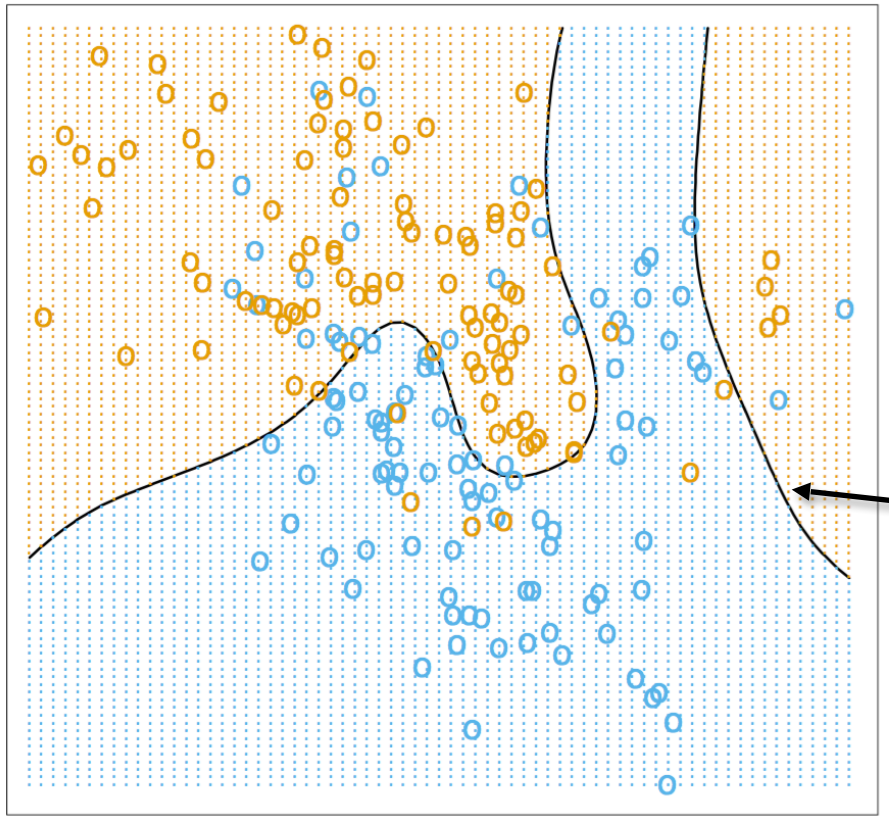
- Assume we have a classification task
- To classify a new point x:
  - Find its k nearest neighbors in the training data
  - Set y to be the majority vote of the labels of these nearest neighbors

What do we mean by nearest neighbors?

- Design choices / hyperparameters:
  - Number of nearest neighbors k
  - Distance metric
  - Aggregation method

Smallest distance in feature space (with d features)

Example: Bayes classifier



Training data:

- True label: +1
- True label: -1

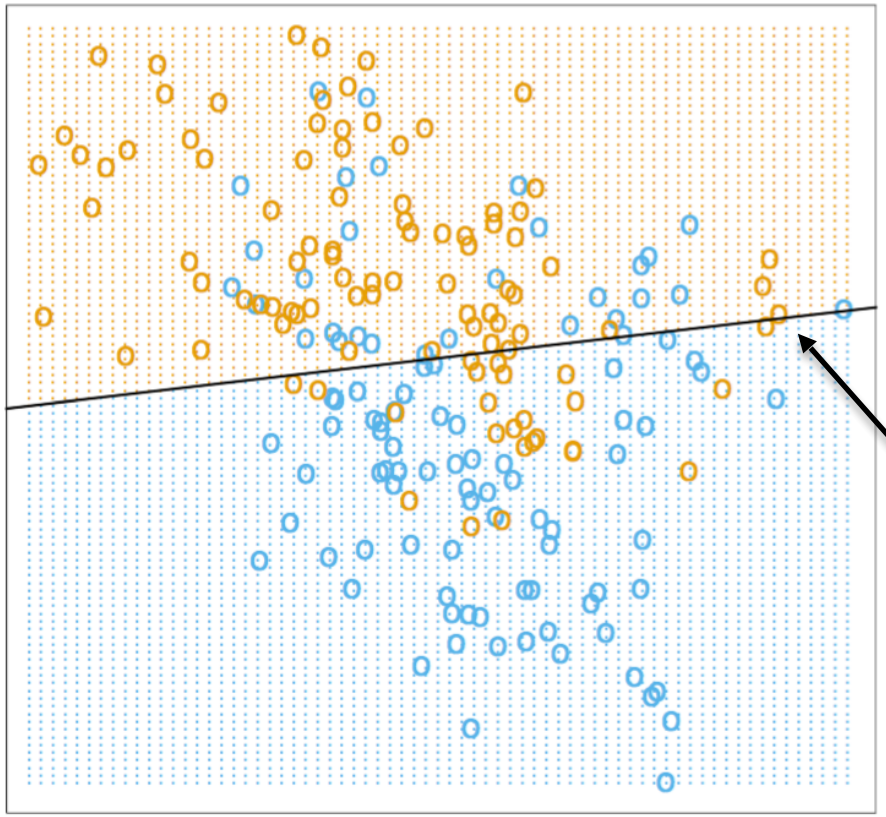
Optimal Bayes classifier:

$$\mathbb{P}(Y = 1|X = x) = \frac{1}{2}$$ # Ground truth

- Predicted label: +1
- Predicted label: -1

Figures from Hastie et al.

Linear decision boundary



Training data:

True label: +1

True label: -1

Learned linear decision boundary:

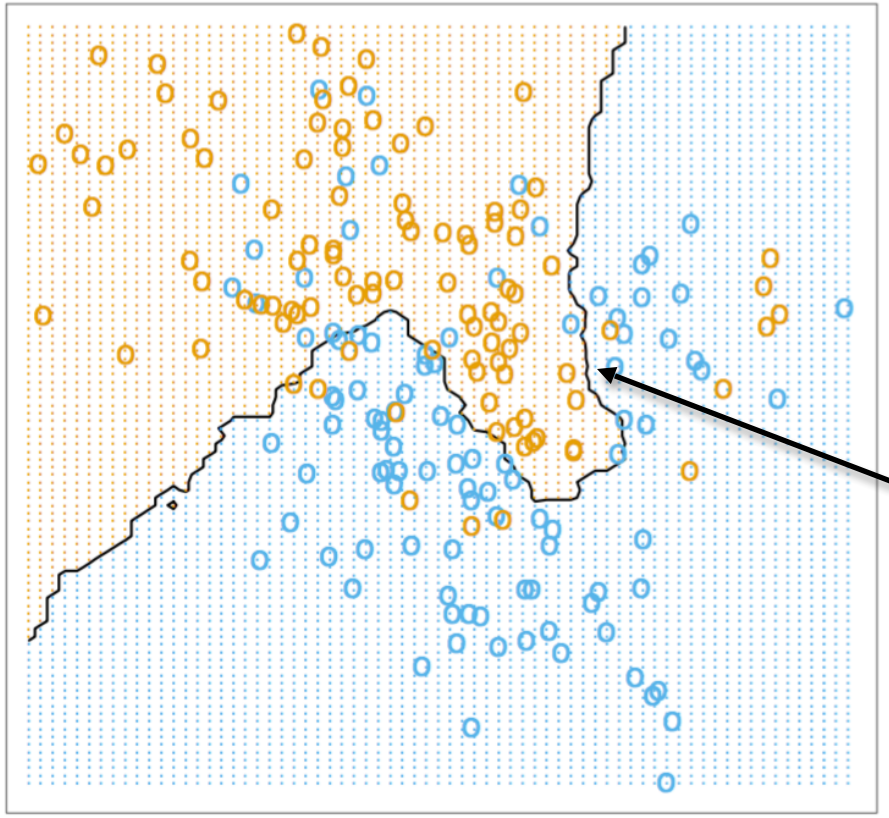
$$x^T w + b = 0$$

Predicted label: +1

Predicted label: -1

Figures from Hastie et al.

$k = 15$ nearest neighbors boundary



Boundary depends on the training data points

Training data:

True label: +1

True label: -1

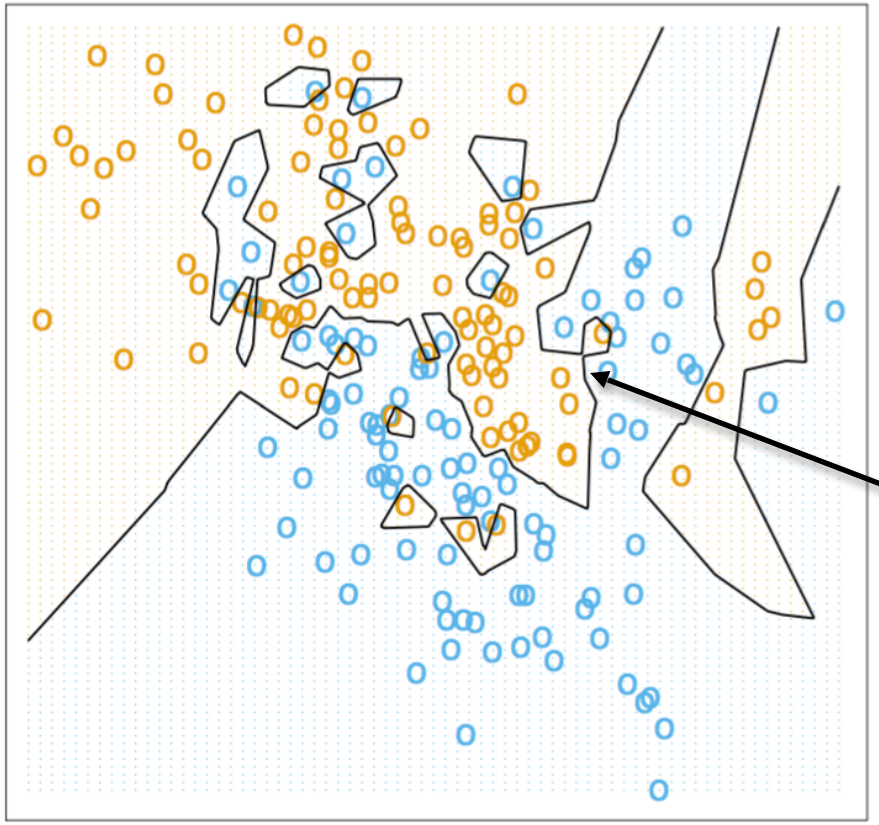
15 nearest neighbors decision boundary (majority vote)

Predicted label: +1

Predicted label: -1

Figures from Hastie et al.

$k = 1$ nearest neighbor boundary



Overfitting

Training data:

True label: +1

True label: -1

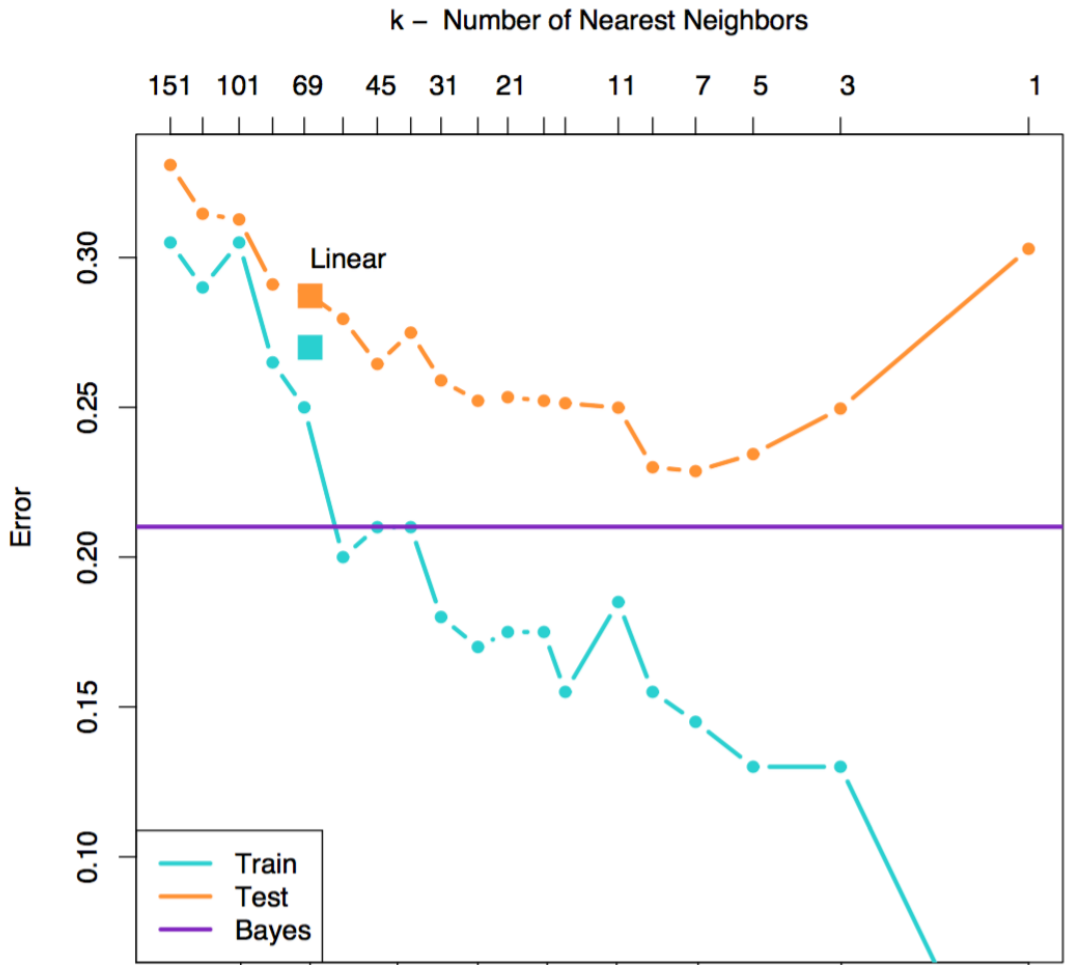
1 nearest neighbor decision boundary (majority vote)

Predicted label: +1

Predicted label: -1

Figures from Hastie et al.

k nearest neighbors error



Underfitting

Overfitting

Parametric vs non-parametric

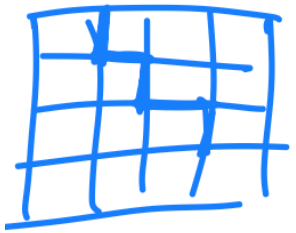
- A model is parametric if # parameters does not depend on # samples

  # After training, you can discard your data

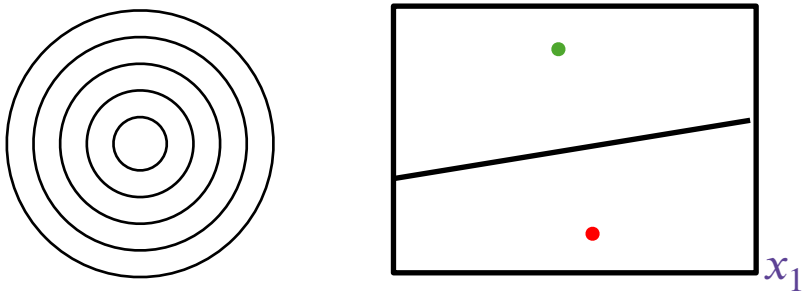
- A model is non-parametric if # parameters increases with # samples

  # Keep training data around for inference

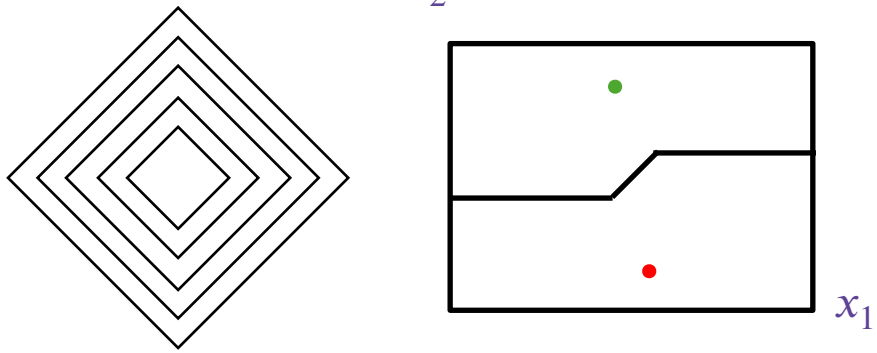
Notable distance metrics & level sets



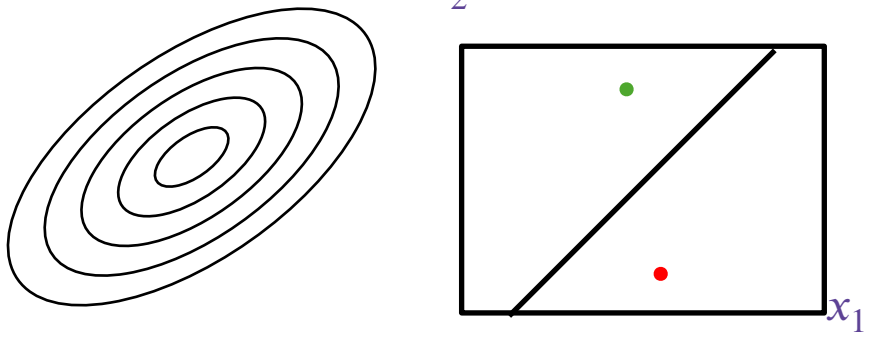
ℓ_2 norm (Euclidean) $d(u,v) = \|u - v\|_2^2$



ℓ_1 norm (Manhattan, taxicab) $\|u - v\|_1 = \sum_{i=1}^{d} |u_i - v_i|$

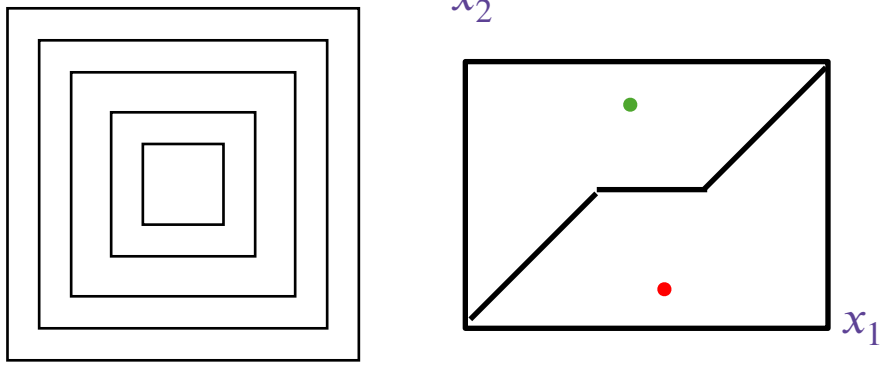


Mahalanobis norm $(u - v)^T M (u - v)$



Weight different dimensions differently

ℓ_∞ norm (max) $\|u - v\|_\infty = \max_i |u_i - v_i|$

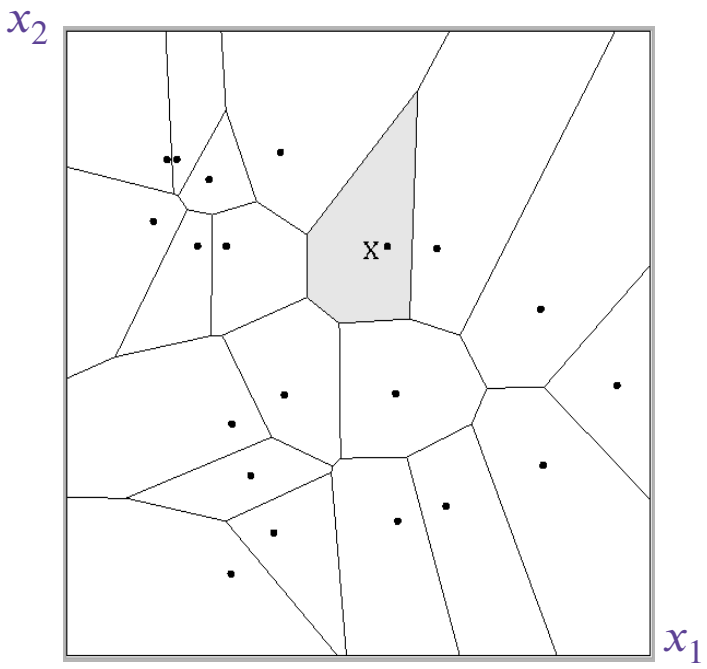


Max distance between vectors

Example: distance metrics with $k = 1$ NN

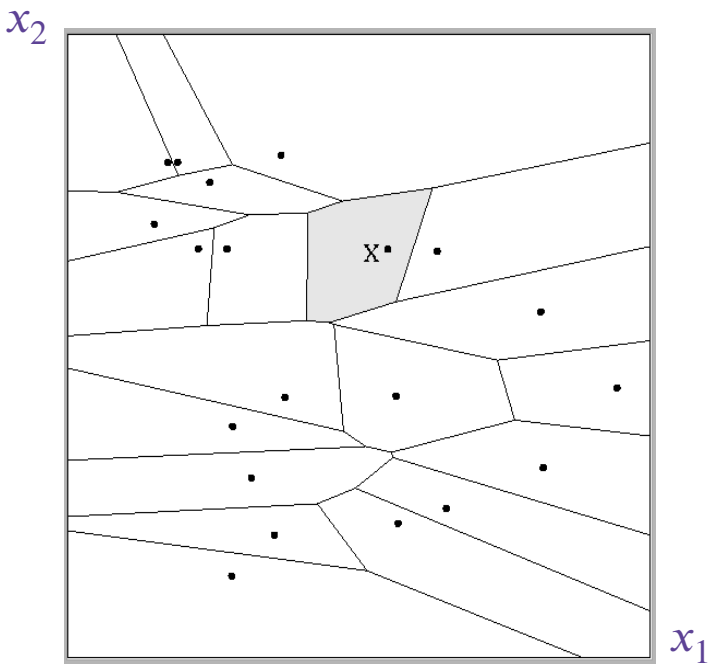
L2

$$d(x, x') = (x_1 - x'_1)^2 + (x_2 - x'_2)^2$$



Mahalanobis

$$d(x, x') = (x_1 - x'_1)^2 + 9(x_2 - x'_2)^2$$



Know the geometry of your feature space

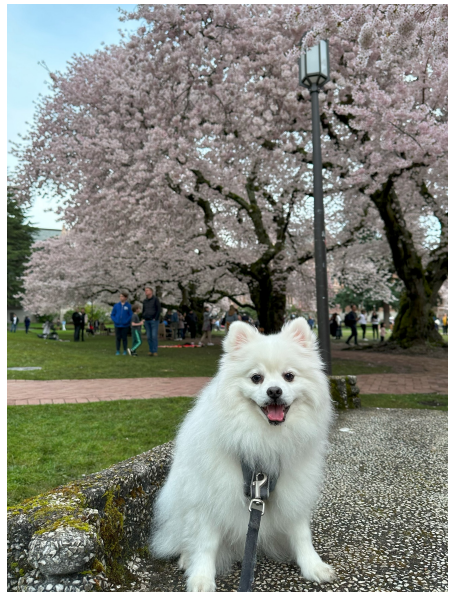
Learned distance metrics

Training data

Dog

Cat

Test data



Use a neural net to learn the distance function

kNN demo

- http://vision.stanford.edu/teaching/cs231n-demos/knn/

1-NN classification: Theoretical guarantees

$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n} \sim P \qquad x^{(i)} \in \mathbb{R}^d \qquad y \in \{0, 1\}$

Assume: we have enough data and **our true function is smooth**

Given test point x, let x_{NN} be the nearest neighbour in D

Error if $y_{NN} \neq y$

Case 1: $y_{NN} = 1, y = 0$ w.p. $P(y = 1 \mid x_{NN})P(y = 0 \mid x)$

Case 2: $y_{NN} = 0, y = 1$ w.p. $P(y = 0 \mid x_{NN})P(y = 1 \mid x)$

As $n \to \infty$,

$P(y \mid x_{NN}) \to P(y \mid x)$ # Why?

By assumptions, distance between x and $x_{NN} \to 0$ as $n \to \infty$

Error: $2P(y = 1 \mid x)P(y = 0 \mid x)$

$= 2p^*(1 - p^*) \quad \leq 2p^*$

Define Bayes error:

$p^* = \min\{P(y = 1 \mid x), P(y = 0 \mid x)\}$

1-NN classification: Theoretical guarantees

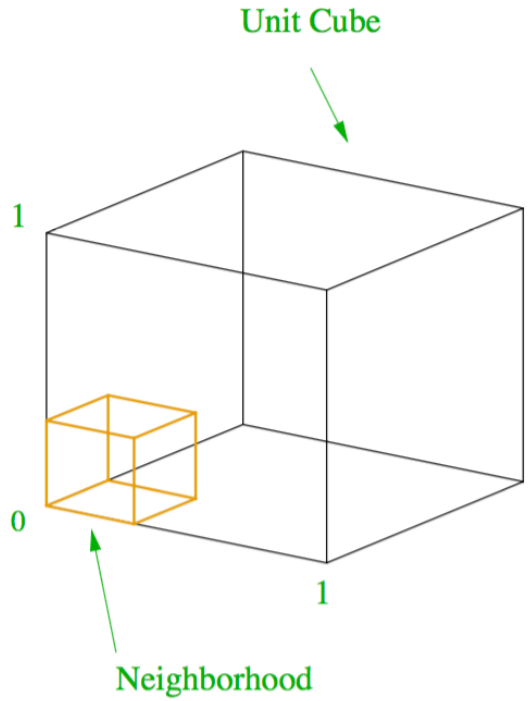
As $n \to \infty$, Error $= 2p^*(1 - p^*) \leq 2p^*$

Theorem[Cover, Hart, 1967] If P_X is supported everywhere in $\mathbb{R}^d$ and $P(Y = 1|X = x)$ is smooth everywhere, then as $n \to \infty$ the 1-NN classification rule has error at most twice the Bayes error rate.

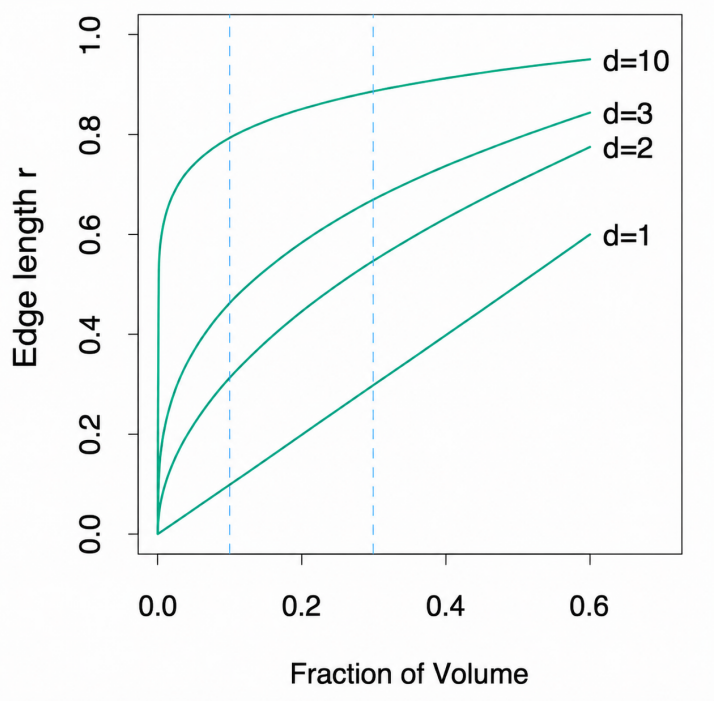
Bayes error is the best you can do given measurement error, so with infinite data nearest neighbors is a really good thing to do!

Can fit arbitrarily complicated functions

Curse of dimensionality, example 1

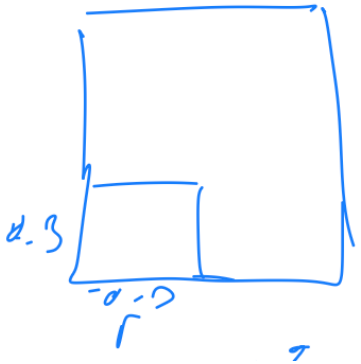


side length r



If d = 1 $P = 0.3$

If d = 2 $P = 0.3^2 = 0.09$



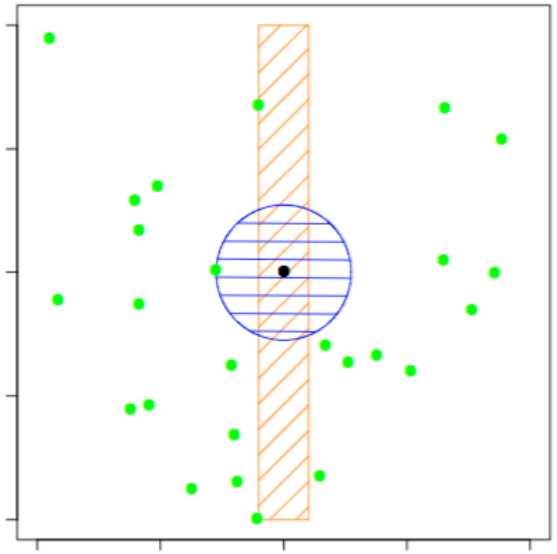
X is uniformly distributed over [0,1]^d. What is $P(X \in [0, r]^d)$**?** $= \frac{1}{r^d}$

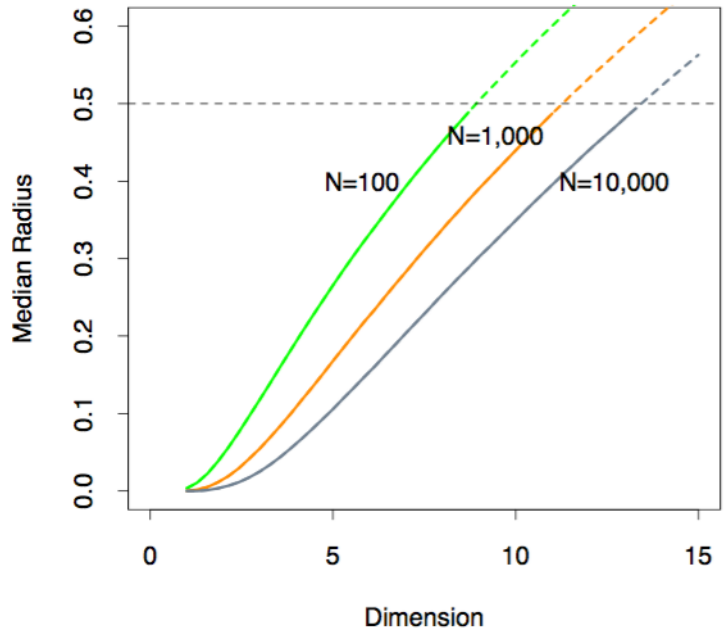
How many samples do we need so that a nearest neighbor is within a cube of side length r**?**

Increases exponentially with dimension p!

Curse of dimensionality, example 2

$\{X_i\}_{i=1}^n$ are uniformly distributed over $[-.5, .5]^p$.



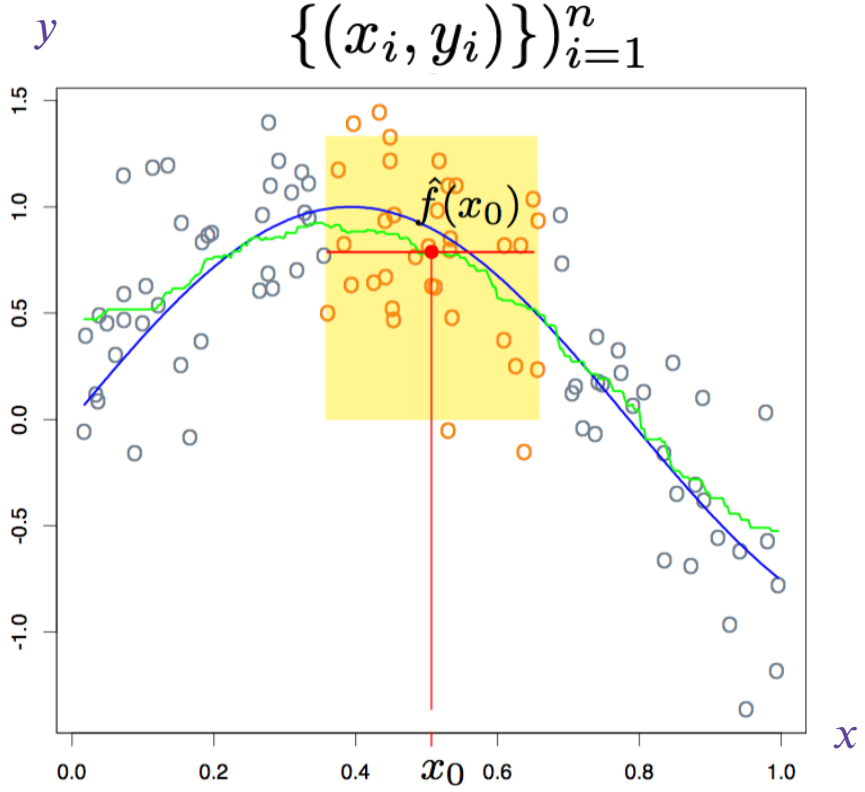


High dimensional space is hard to cover

What is the median distance from a point at origin to its 1NN?

How many samples do we need so that a median Euclidean distance is within r?

Nearest neighbor regression



kNN regressor:

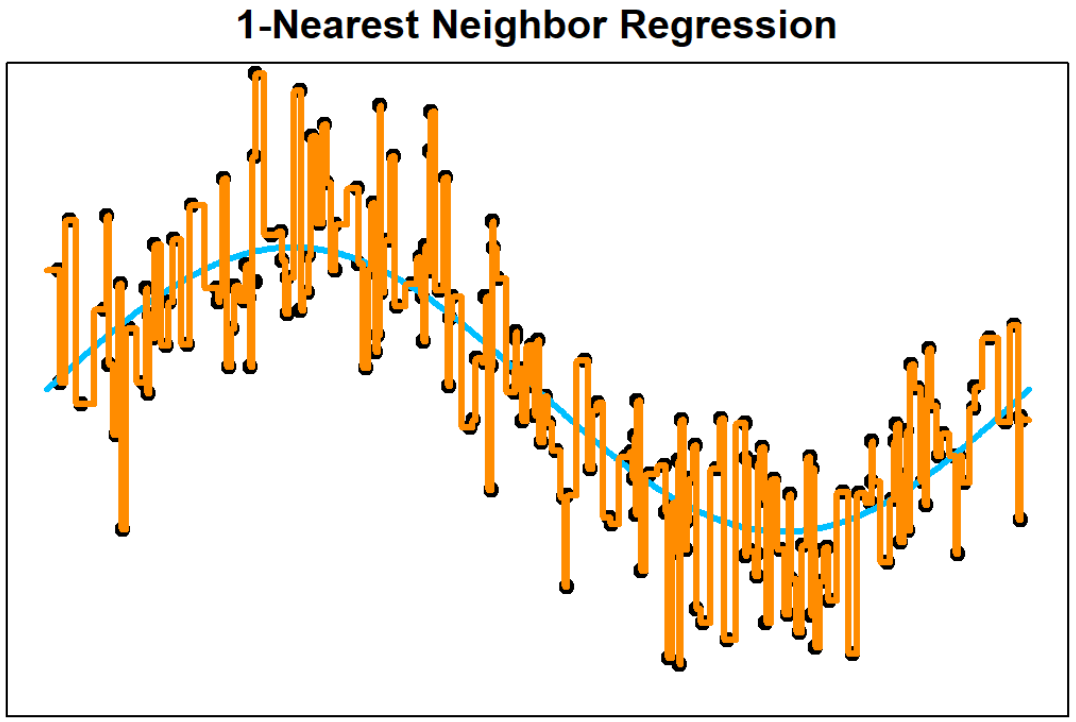
$$\hat{f}(x) = \frac{1}{k} \sum_{i \in kNN(x)} y^{(i)}$$

$$= \frac{1}{k} \sum_{i=1}^{n} y^{(i)} 1\{x^{(i)} \in kNN(x)\}$$

Recall: $f^*(x) = \arg\min_f \mathbb{E}[(f(x) - y)^2]$

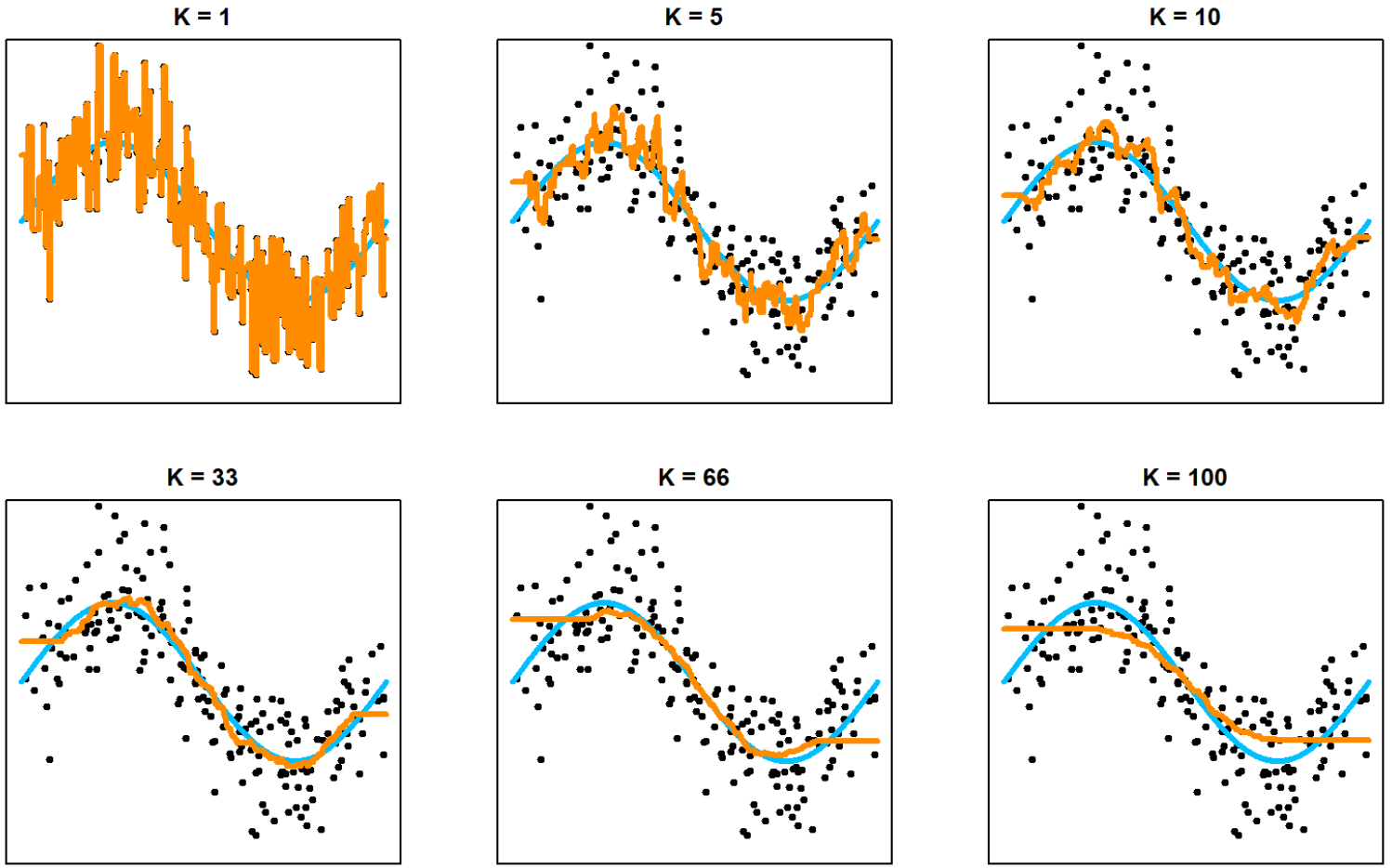
$$= \mathbb{E}[y \mid x]$$

Overfitting



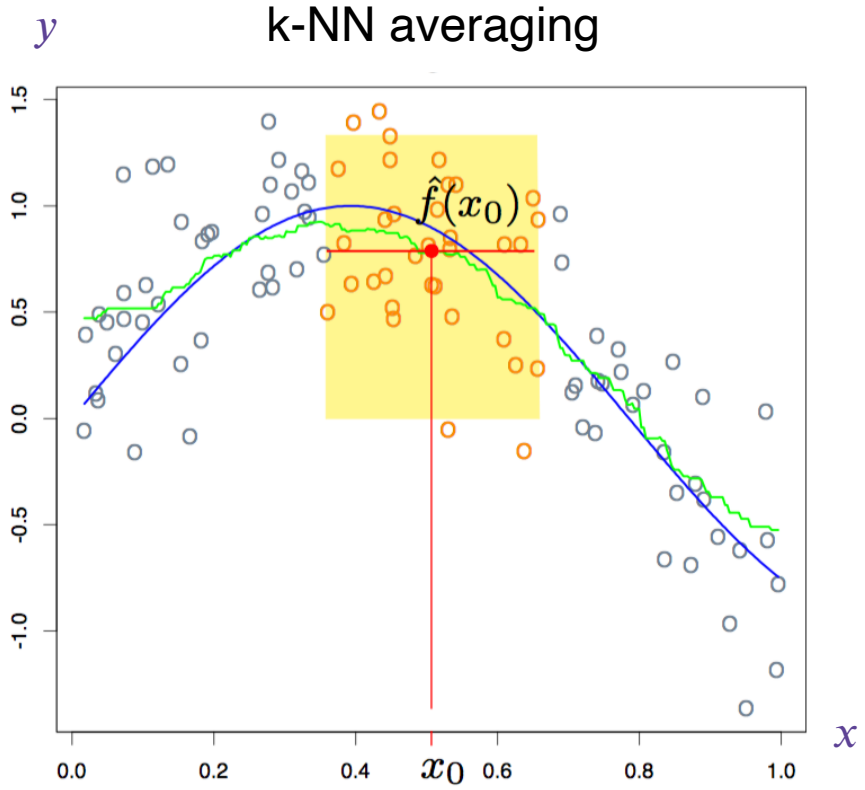
Figures from Ruoqing Zhu, Stat 542

Bias vs variance

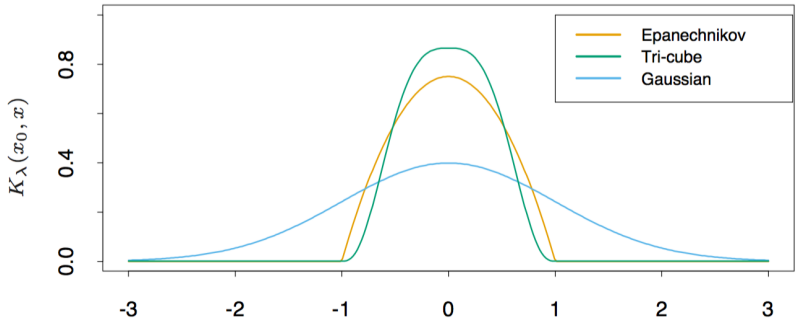


Figures from Ruoqing Zhu, Stat 542

Smoothed nearest neighbor regression



smoothing: $K(x, y)$

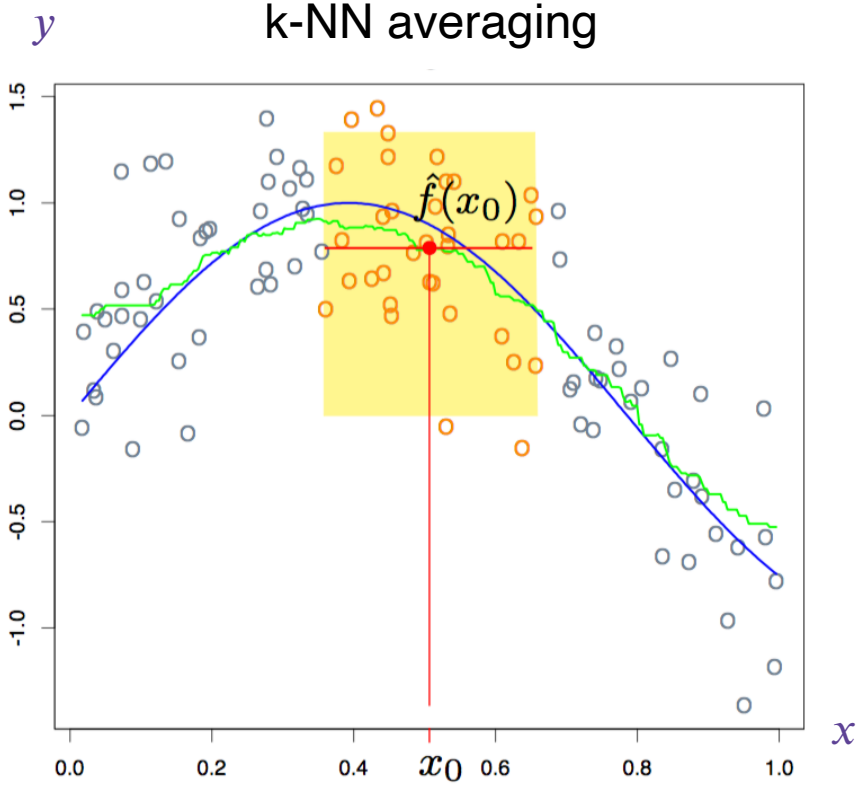


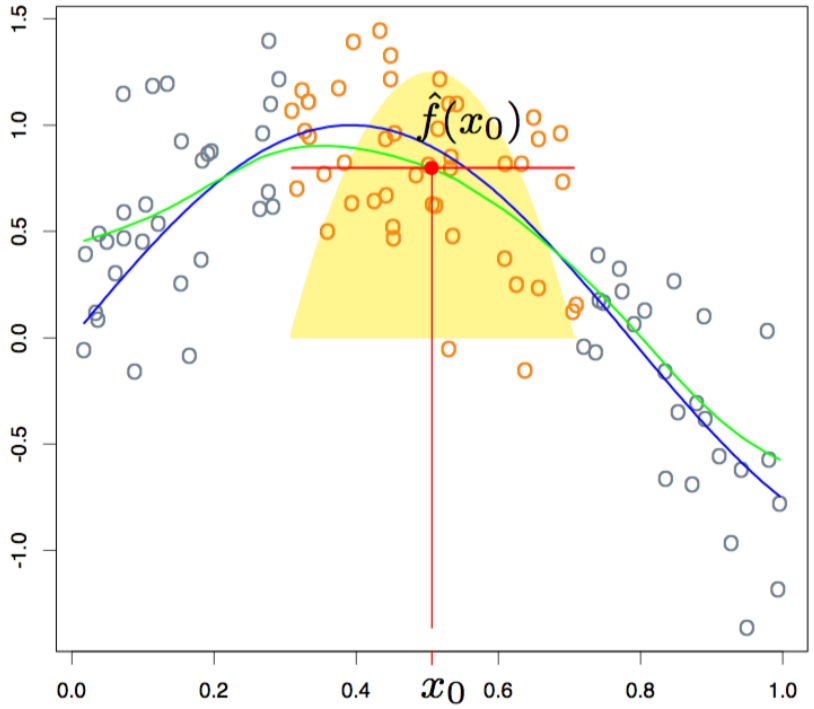
$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K\left(x, x^{(i)}\right) y^{(i)}}{\sum_{i=1}^{n} K\left(x, x^{(i)}\right)}$$

Smoothed nearest neighbor regression

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K\left(x, x^{(i)}\right) y^{(i)}}{\sum_{i=1}^{n} K\left(x, x^{(i)}\right)}$$

k-NN averaging

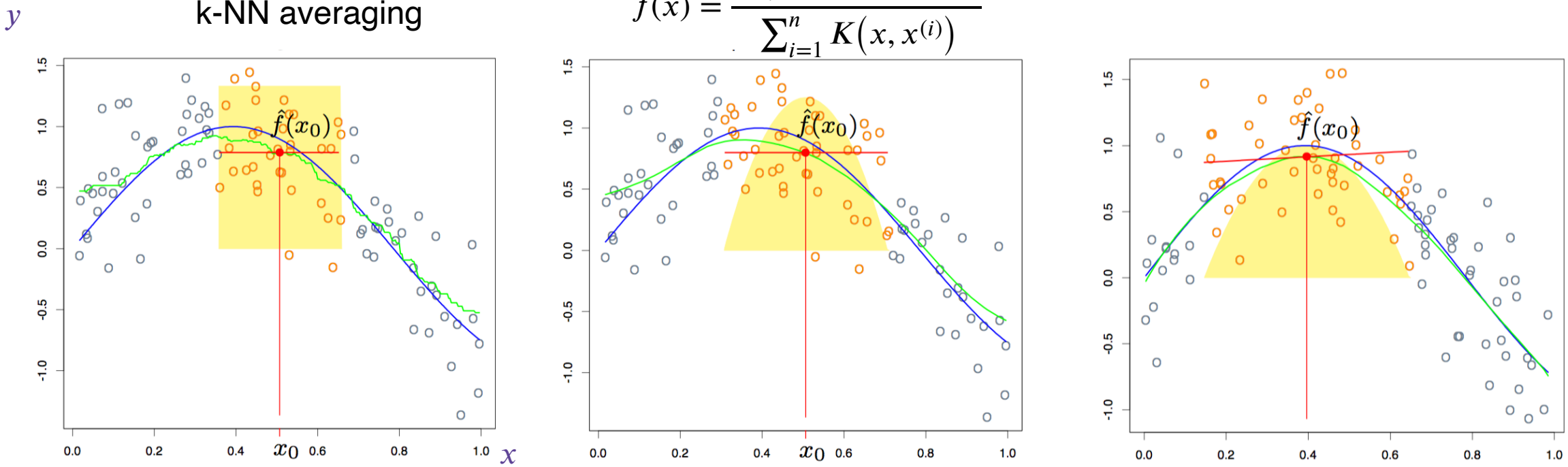




Locally linear regression

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K\left(x, x^{(i)}\right) y^{(i)}}{\sum_{i=1}^{n} K\left(x, x^{(i)}\right)}$$

k-NN averaging



Foreshadowing kernels

- Kernel methods are non-parametric:

$$f(x) = \sum_{i=1}^{n} \alpha_i K\left(x^{(i)}, x\right)$$

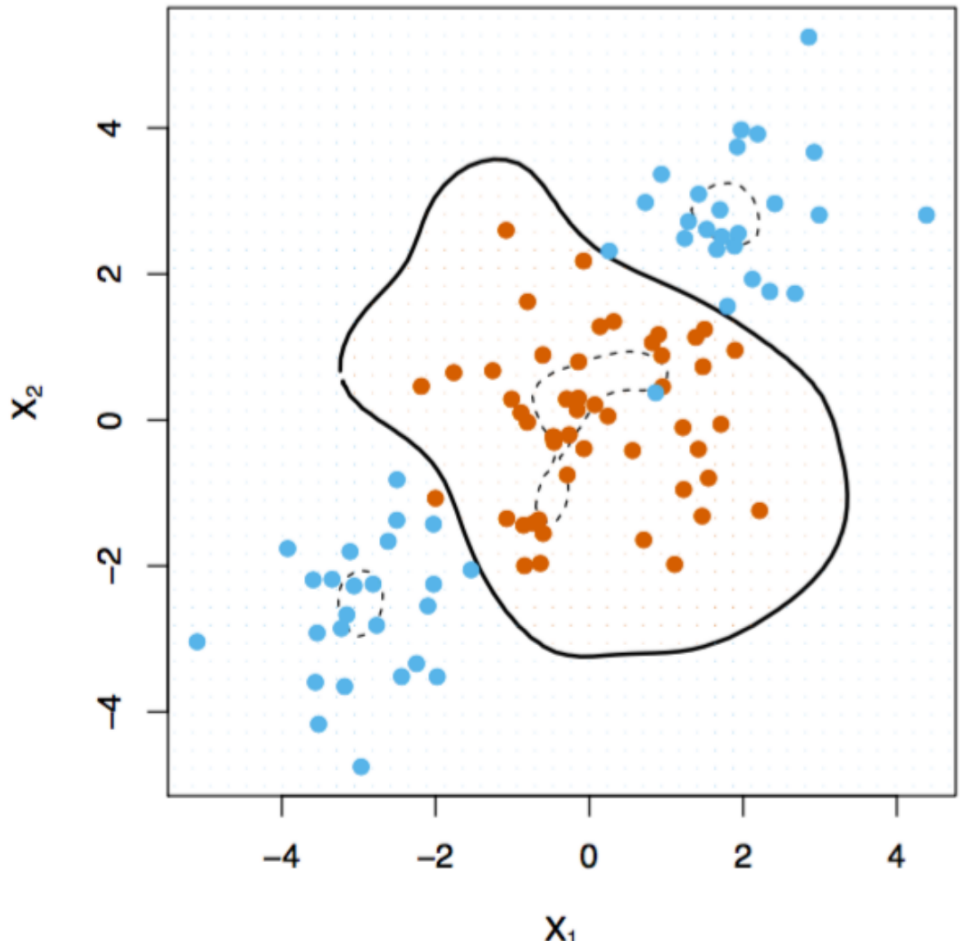
  # parameters goes up with # data

- Compare with (smoothed) nearest neighbors:

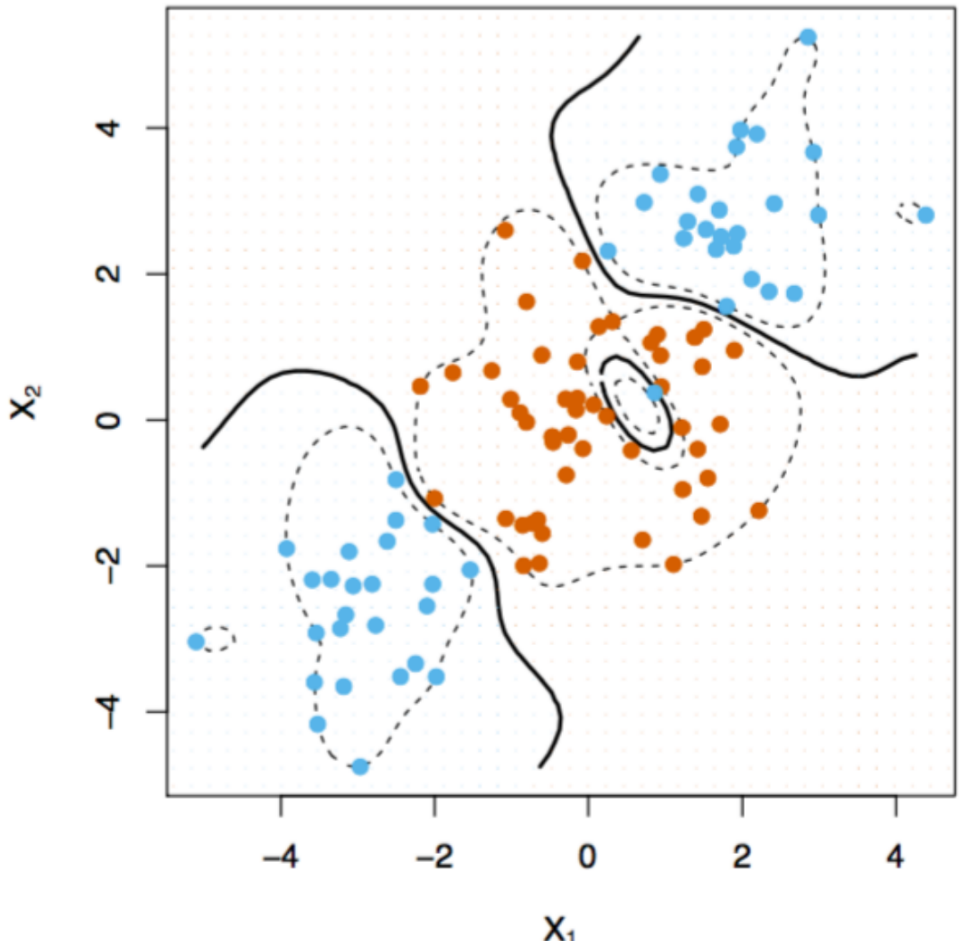
$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K\left(x, x^{(i)}\right) y^{(i)}}{\sum_{i=1}^{n} K\left(x, x^{(i)}\right)}$$

The Radial Basis Function (RBF) kernel $\exp\left(-\frac{\left\|x^{(i)}-x\right\|_2^2}{2\sigma^2}\right)$

Bandwidth σ is large enough

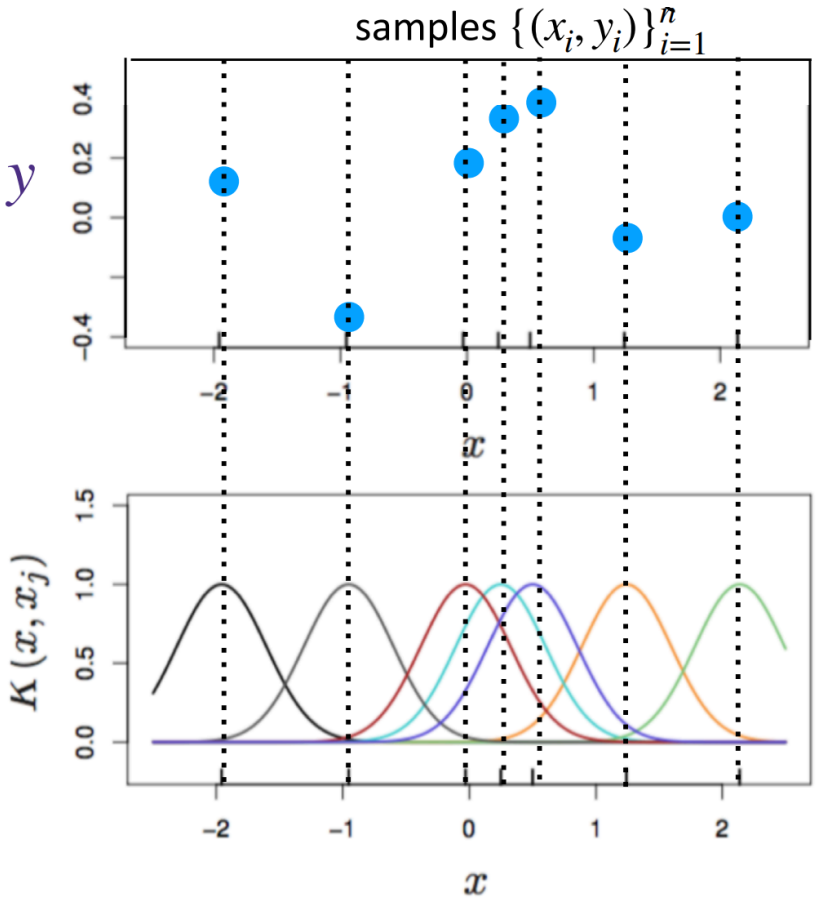


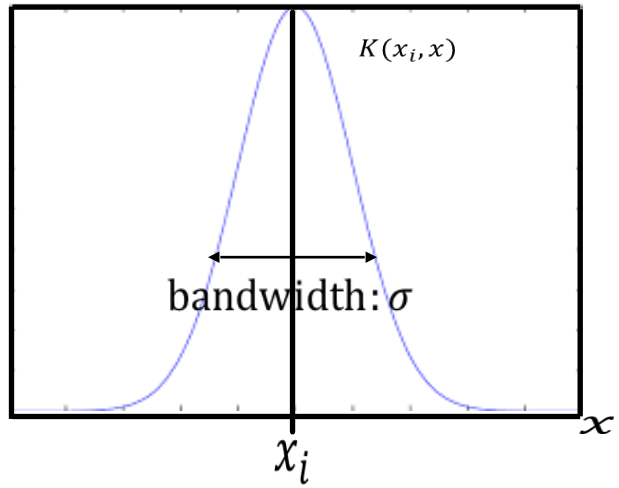
Bandwidth σ is small



The Radial Basis Function (RBF) kernel $\exp\left(-\frac{\left\|x^{(i)}-x\right\|_2^2}{2\sigma^2}\right)$

$$f(x) = \sum_{i=1}^{n} \alpha_i K\left(x^{(i)}, x\right)$$





Kernel methods

$$f(x) = \sum_{i=1}^{n} \alpha_i K\left(x^{(i)}, x\right)$$

- Can be seen as a soft, learned version of “nearest” neighbors
- $K\left(x^{(i)}, x\right) = \phi\left(x^{(i)}\right)^{\top} \phi(x)$ defines “similarity” between $x^{(i)}$ and x
- How many parameters?

Takeaways

- k-NN is very simple to explain and implement
- No training! But inference can still be computationally demanding.
- You can use other forms of distance (not just Euclidean)
- Smoothing and local linear regression can improve performance (at the cost of higher variance)
- With a lot of data, “local methods” have strong, simple theoretical guarantees
- Without a lot of data, neighborhoods aren’t “local” and methods suffer (curse of dimensionality)