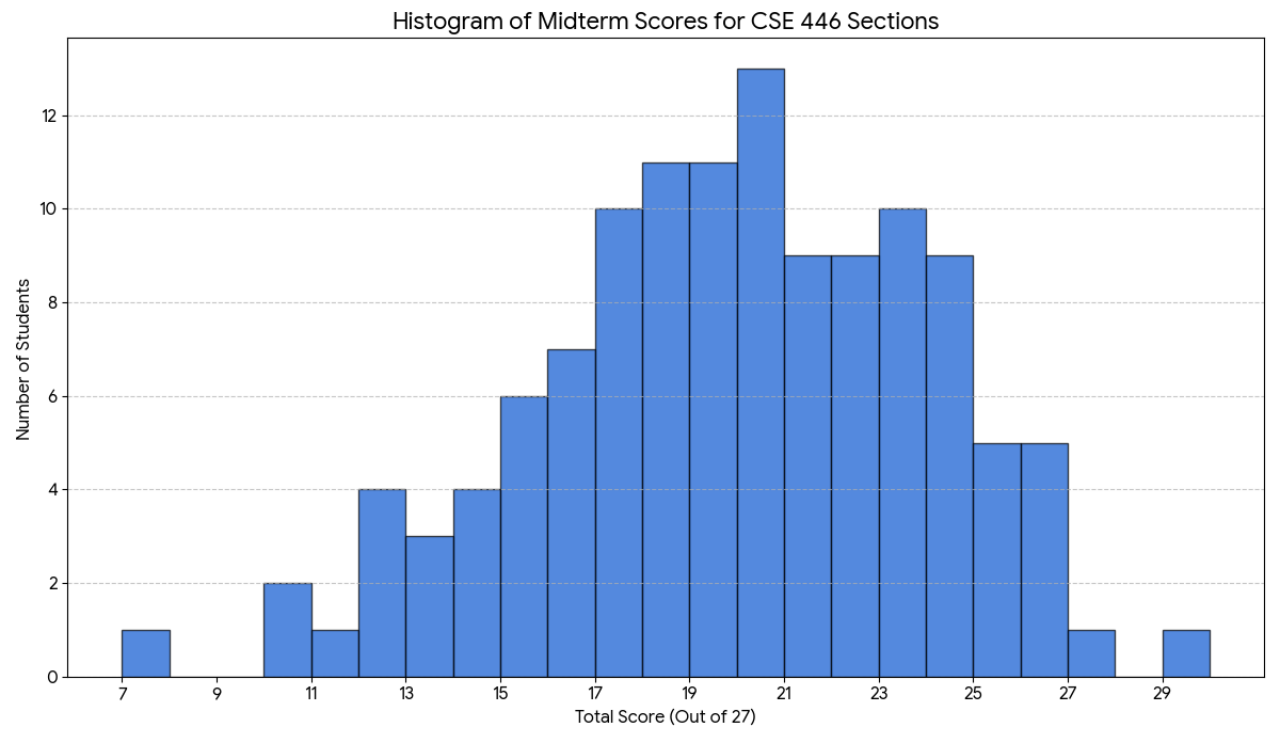


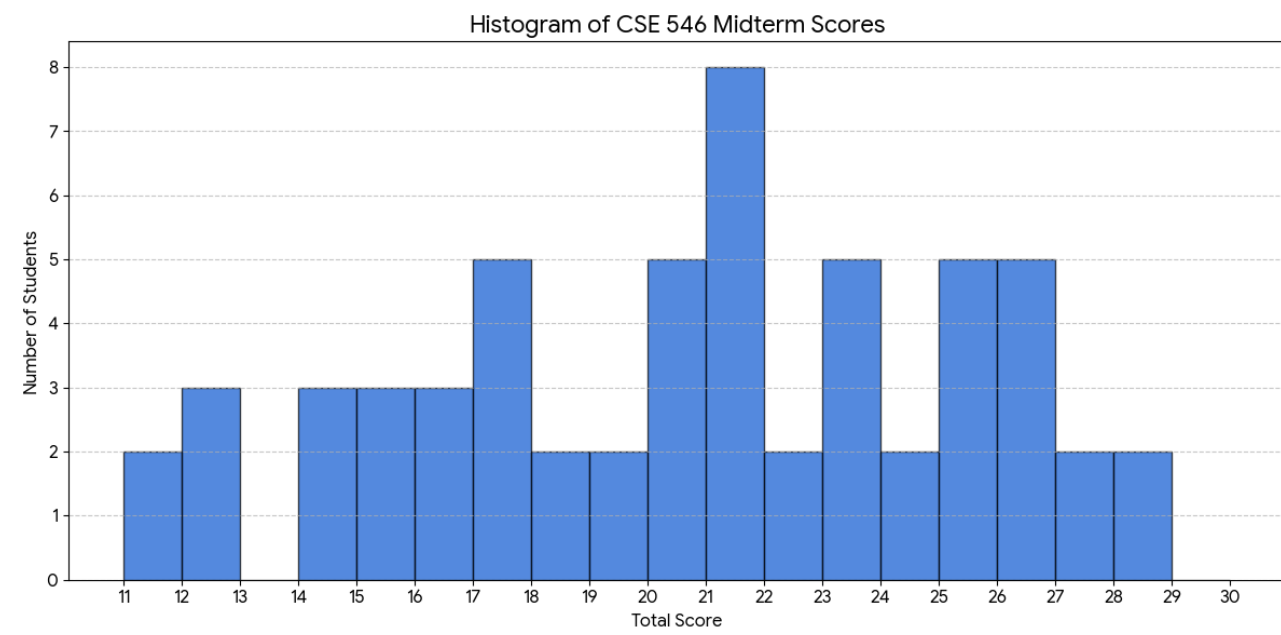
Midterm results: CSE 446

- Median: 20.0 / 27
- Standard deviation: 4.14



Midterm results: CSE 546

- Median: 21.4 / 27
- Standard deviation: 4.64



Midterm results

- Exam scores are imperfect metrics
- Midterm is 20% of course grade – you can still do well
- You can submit regrade requests 11/12 @ 11:59PM
- Clarifications? Come to OH or post on Ed

Q7

7. 1 point

Give a value for the constant $c \in \mathbb{R}$ that makes f a valid PDF:

$$f(x) = \begin{cases} cx, & \text{if } 0 < x < 10 \\ 0, & \text{otherwise} \end{cases}$$

Q9

9. 1 point

Assume we have n training data points sampled from some population distribution, which we then use to train a model. Is leave-one-out cross validation (LOOCV) error (1) typically a slight underestimate, (2) an unbiased estimator, or (3) typically a slight overestimate of the true (population) error of the final model trained on all n data points? Briefly explain your answer.

Q11

11. Consider the following linear regression setting. Let $X \in \mathbb{R}^{n \times d}$, $\lambda \geq 0$, and suppose $y \in \mathbb{R}^n$ is drawn from a Gaussian, $y \sim \mathcal{N}(Xw, \sigma^2 I)$ for some $w \in \mathbb{R}^d$, $\sigma^2 > 0$. Let

$$\hat{w} = \arg \min_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \lambda \|w\|_2^2.$$

(a)

If $n < d$ and $\lambda = 0$, how many solutions does the above optimization problem have?

Answer: _____

(b)

If $n < d$ and $\lambda > 0$, how many solutions does the above optimization problem have?

Answer: _____

(c)

Fill in the blanks with **increases/decreases/does not change**. As σ^2 **increases**:

The bias of \hat{w} _____

The variance of \hat{w} _____

The irreducible error _____

Q19

19. You train a logistic regression classifier with features $x = (x_1, x_2)$, using

$$P(y = 1|x) = \sigma(w_0 + w_1x_1 + w_2x_2), \quad \text{where } \sigma(z) = \frac{1}{1 + e^{-z}}.$$

a) 1 point

Describe the geometric shape of the decision boundary in this feature space.

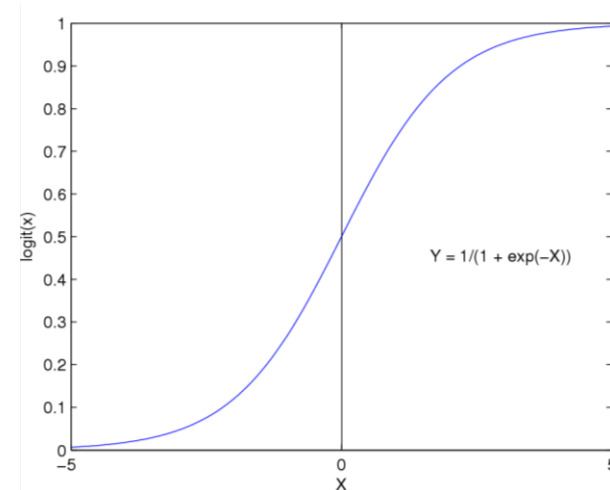
Modeling conditional probabilities

$$\hat{w}_{\text{MLE}} = \operatorname{argmax}_w \sum_{i=1}^n \log P_w(y_i|x_i)$$

- Logistic regression uses a model specialized for (binary) classification:

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-w^T x}}$$

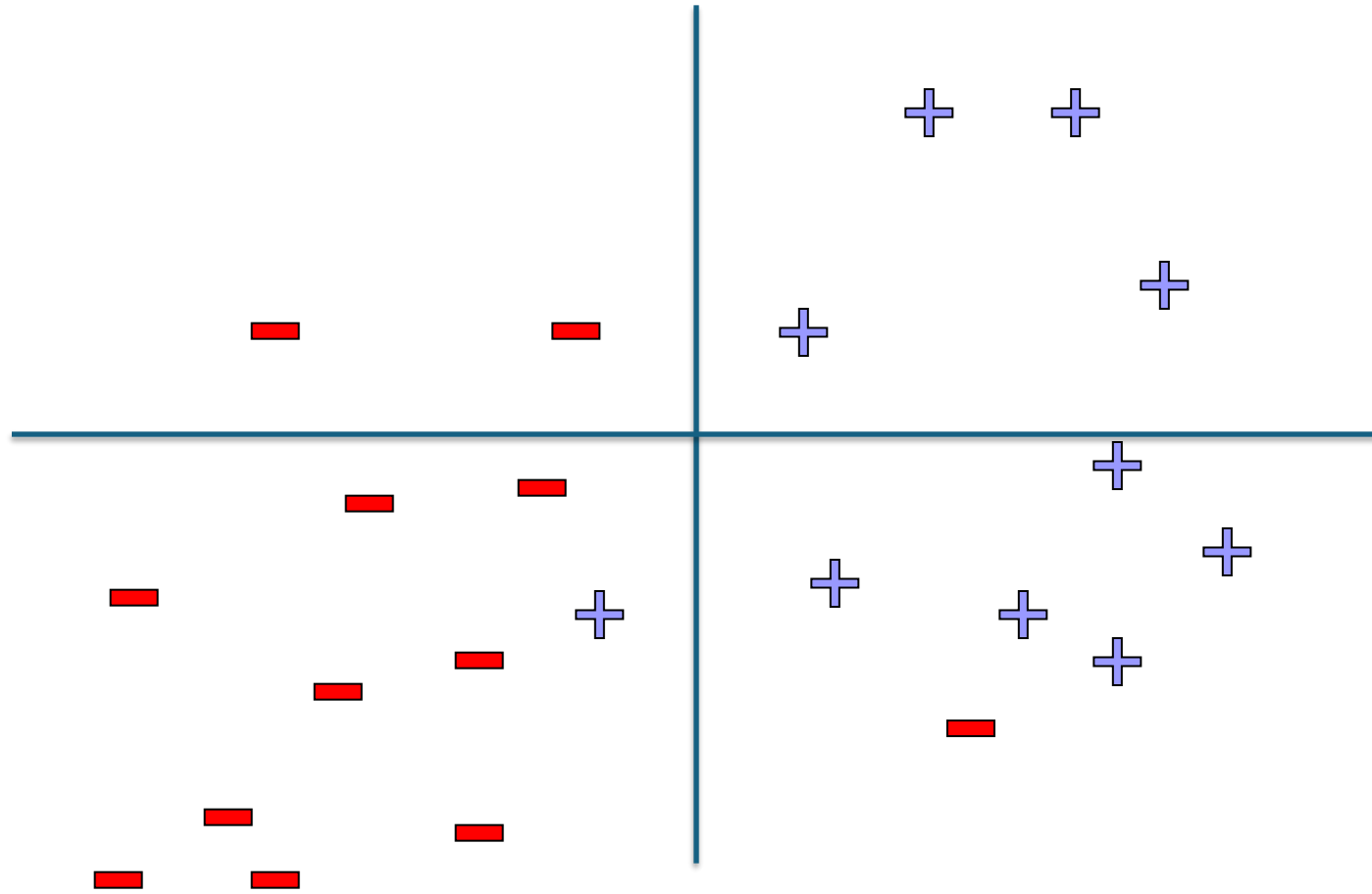
$$P(Y = 0|X = x) = \frac{e^{-w^T x}}{1 + e^{-w^T x}}$$



Sigmoid for binary classification

- What's the shape of the decision rule $P(Y = 1|X) \geq P(Y = 0|X)$?

Logistic regression – a linear classifier



Q19

19. You train a logistic regression classifier with features $x = (x_1, x_2)$, using

$$P(y = 1|x) = \sigma(w_0 + w_1x_1 + w_2x_2), \quad \text{where } \sigma(z) = \frac{1}{1 + e^{-z}}.$$

a) 1 point

Describe the geometric shape of the decision boundary in this feature space.

21. [Extra credit. Not attempting this question will not affect your grade.] Consider linear regression on 2-dimensional data, where the features of the i -th data point are denoted $(x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$ and the target of the i -th data point is $y^{(i)} \in \mathbb{R}$. There are n training data points, $\{(x_1^{(i)}, x_2^{(i)}), y^{(i)}\}_{i=1}^n$. Assume that the following always holds in the training data, for all $i = 1, 2, \dots, n$:

$$\begin{aligned}x_2^{(i)} &= 2x_1^{(i)} \\ y^{(i)} &= 5x_1^{(i)}.\end{aligned}$$

1. 1 point Suppose we train a ridge regression model, i.e., we solve

$$\hat{w} = \arg \min_{(w_1, w_2) \in \mathbb{R}^2} \sum_{i=1}^n (w_1 x_1^{(i)} + w_2 x_2^{(i)} - y^{(i)})^2 + \lambda(w_1^2 + w_2^2)$$

with $\lambda > 0$. What is \hat{w} in the limit as $\lambda \rightarrow 0$ from above? Hint: It is not necessary to formally take the limit; just assume λ is vanishingly small but positive.

21. [Extra credit. Not attempting this question will not affect your grade.] Consider linear regression on 2-dimensional data, where the features of the i -th data point are denoted $(x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$ and the target of the i -th data point is $y^{(i)} \in \mathbb{R}$. There are n training data points, $\{(x_1^{(i)}, x_2^{(i)}), y^{(i)}\}_{i=1}^n$. Assume that the following always holds in the training data, for all $i = 1, 2, \dots, n$:

$$\begin{aligned}x_2^{(i)} &= 2x_1^{(i)} \\ y^{(i)} &= 5x_1^{(i)}.\end{aligned}$$

2. 1 point Now suppose we train a Lasso regression model, i.e., we solve

$$\hat{w} = \arg \min_{(w_1, w_2) \in \mathbb{R}^2} \sum_{i=1}^n (w_1 x_1^{(i)} + w_2 x_2^{(i)} - y^{(i)})^2 + \lambda(|w_1| + |w_2|)$$

with $\lambda > 0$. What is \hat{w} in the limit as $\lambda \rightarrow 0$ from above? Hint: It is not necessary to formally take the limit; just assume λ is vanishingly small but positive.

Non-parametric methods: Nearest neighbors

CSE 446/546

Sewoong Oh & Pang Wei Koh

Parametric vs non-parametric

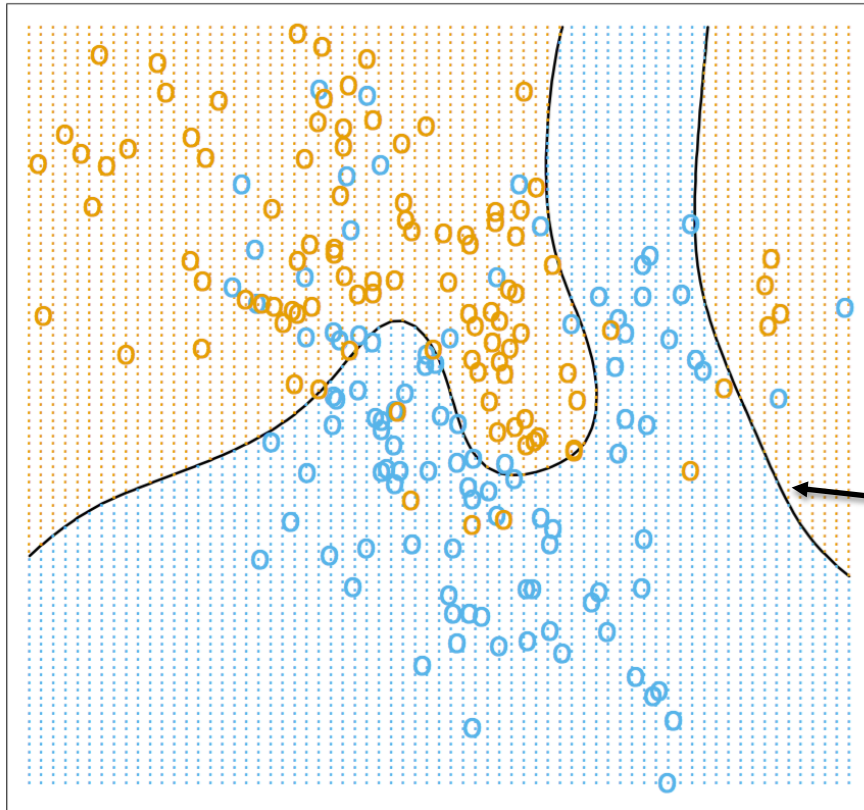
- A model is parametric if # parameters does not depend on # samples
- A model is non-parametric if # parameters increases with # samples
 - Does not mean absence of parameters!

Why non-parametric models?

This lecture: k nearest neighbors

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

Example: Bayes classifier



Training data:

○ True label: +1

○ True label: -1

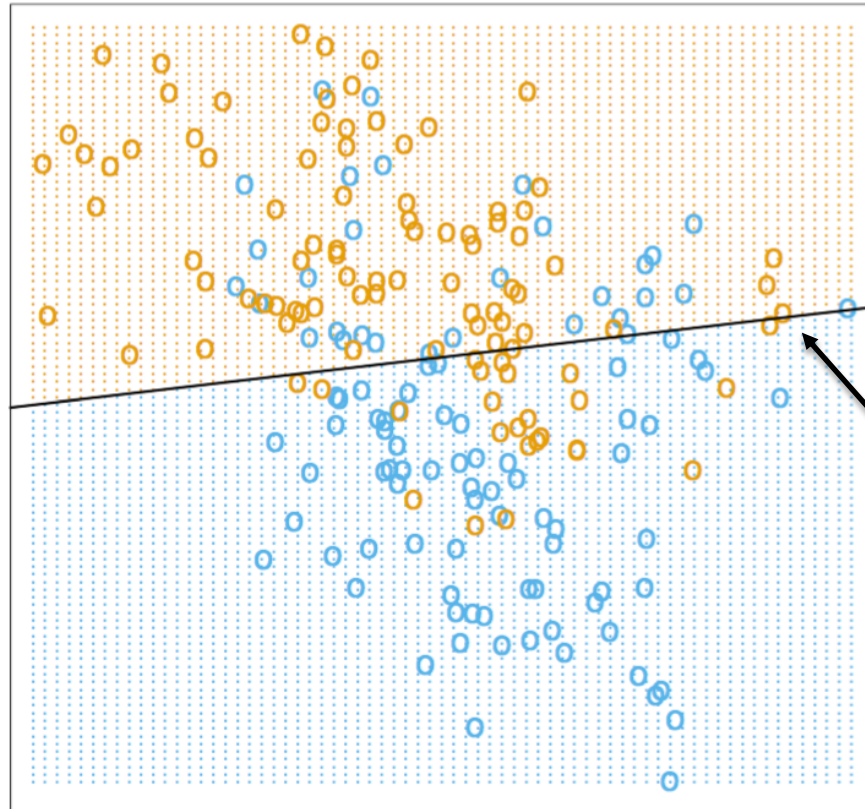
Optimal Bayes classifier:

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

▨ Predicted label: +1

▨ Predicted label: -1

Linear decision boundary



Training data:

○ True label: +1

○ True label: -1

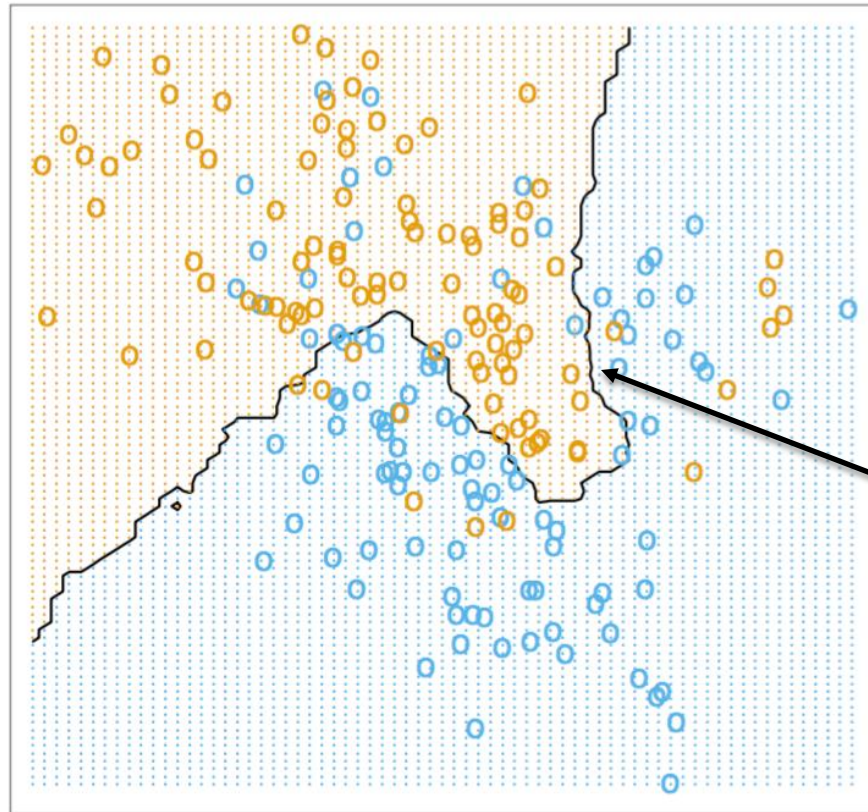
Learned linear decision boundary:

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

▨ Predicted label: +1

▨ Predicted label: -1

$k = 15$ nearest neighbors boundary



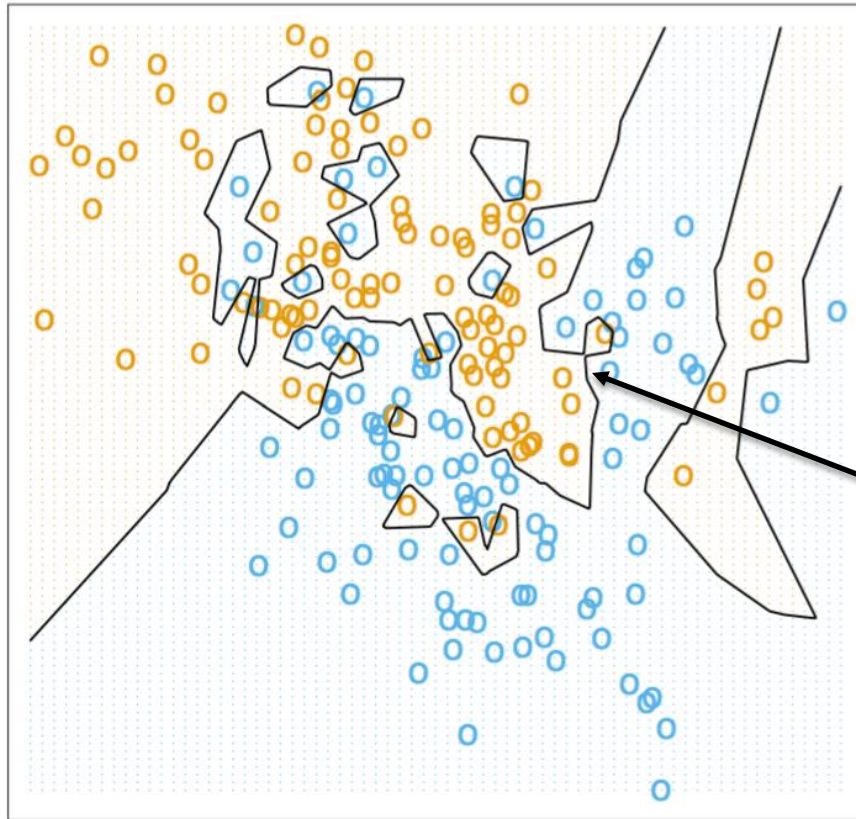
Training data:

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

15 nearest neighbors decision boundary (majority vote)

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

$k = 1$ nearest neighbor boundary



Training data:

○ True label: +1

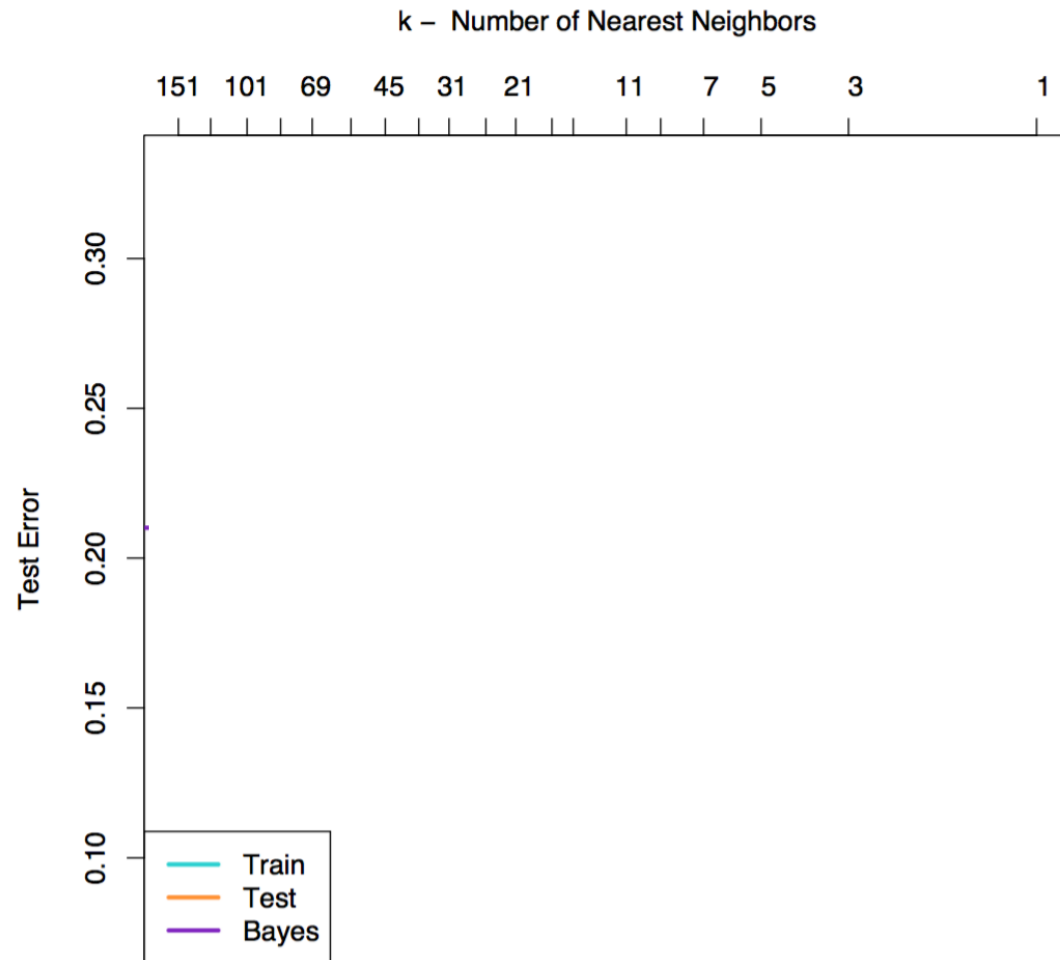
○ True label: -1

1 nearest neighbor decision boundary (majority vote)

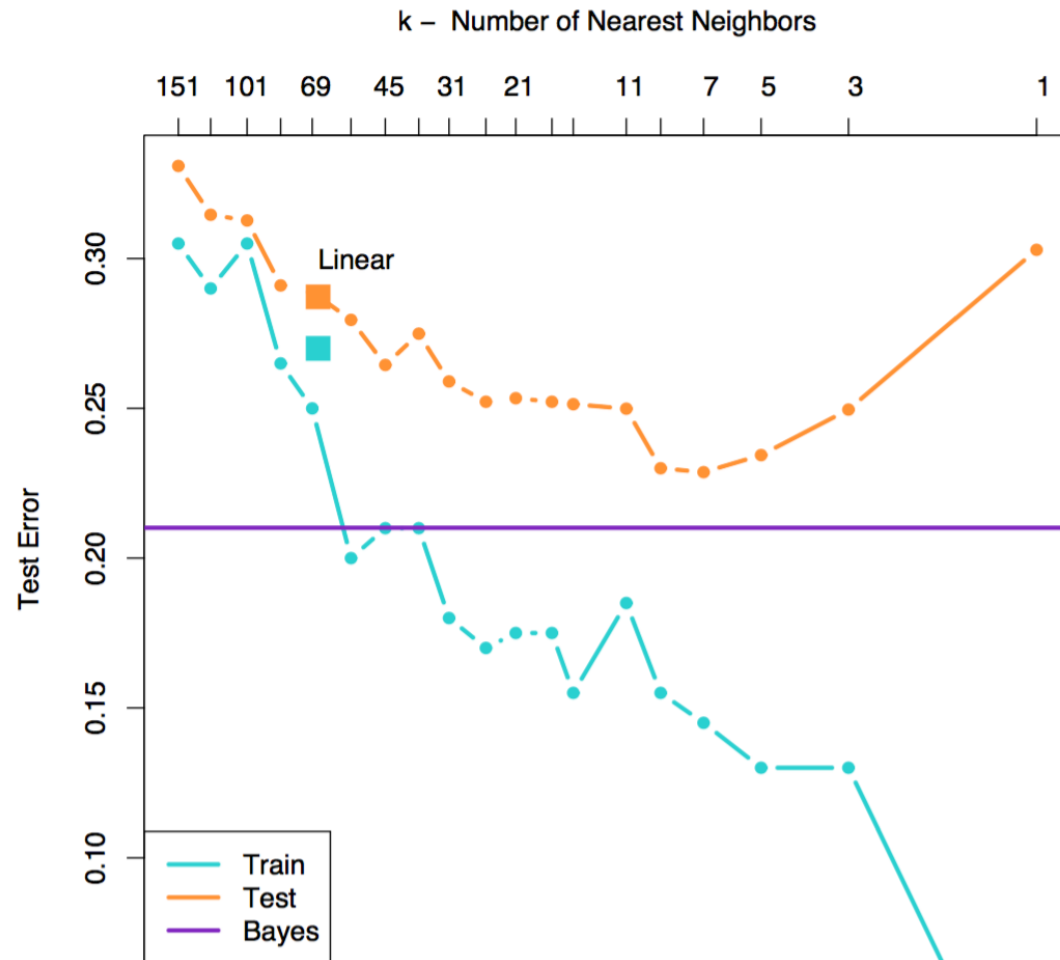
■ Predicted label: +1

■ Predicted label: -1

k nearest neighbors error



k nearest neighbors error

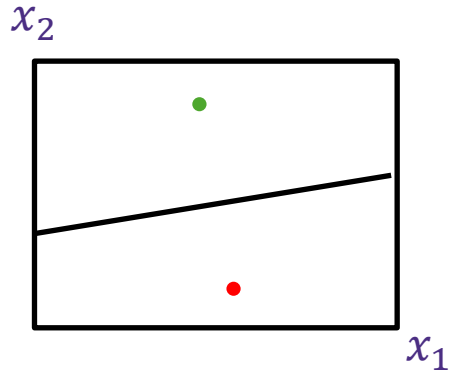
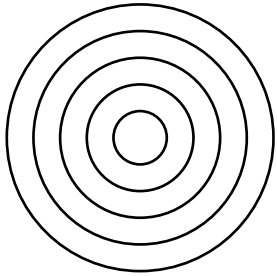


Parametric vs non-parametric

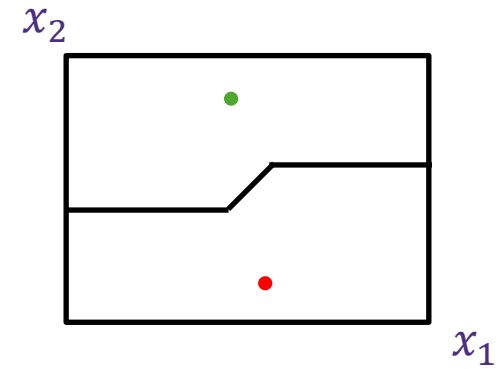
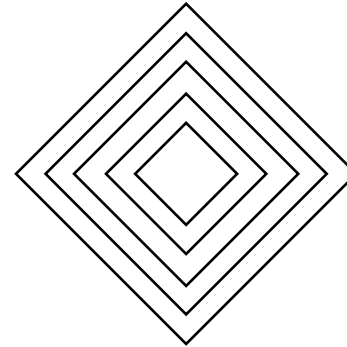
- A model is parametric if # parameters does not depend on # samples
- A model is non-parametric if # parameters increases with # samples

Notable distance metrics & level sets

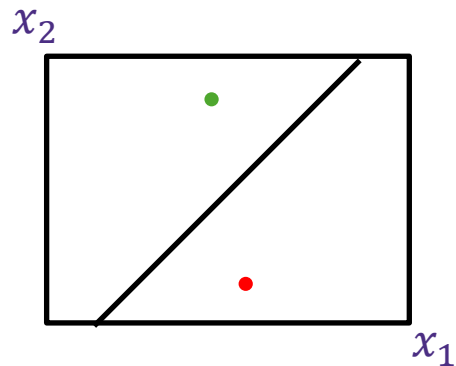
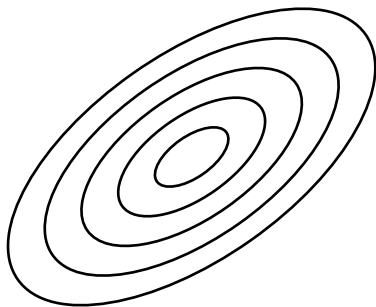
ℓ_2 norm (Euclidean)



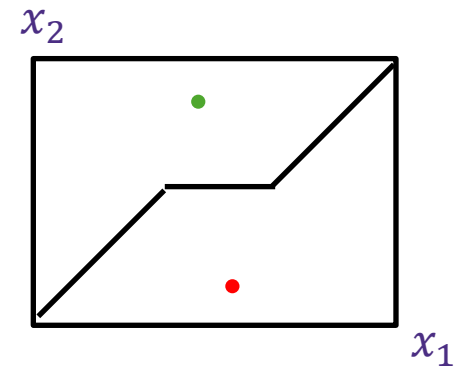
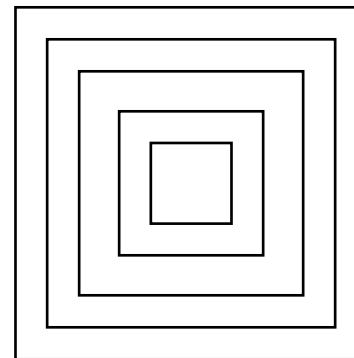
ℓ_1 norm (Manhattan, taxicab)



Mahalanobis norm

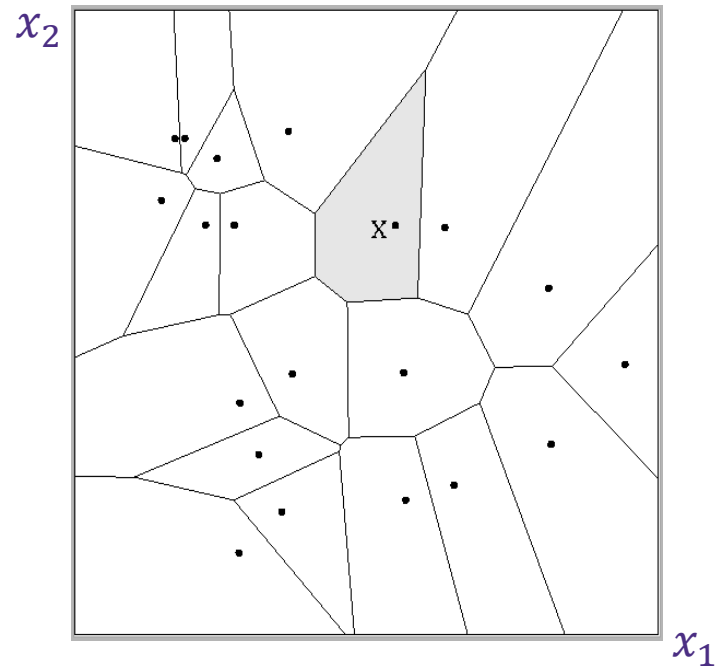


ℓ_∞ norm (max)

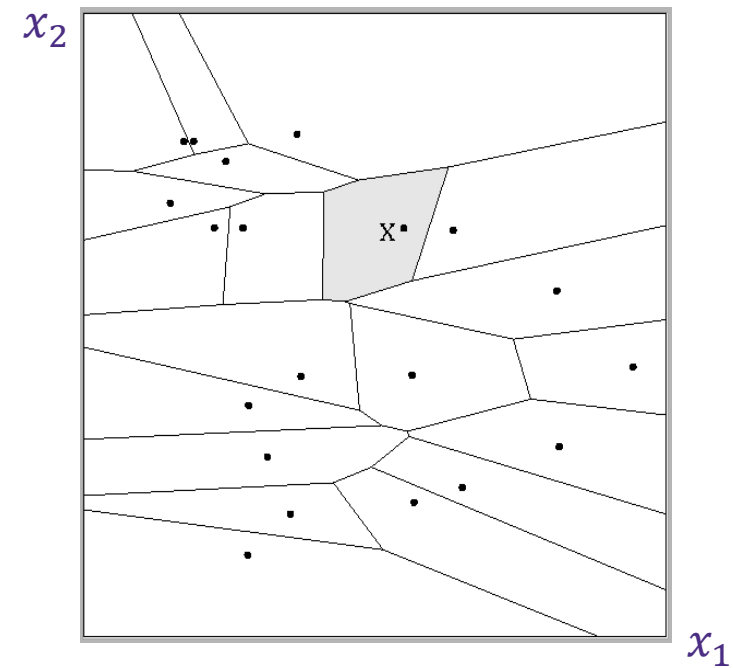


Example: distance metrics with $k = 1$ NN

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



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



Learned distance metrics

Training data



Dog



Cat

Test data

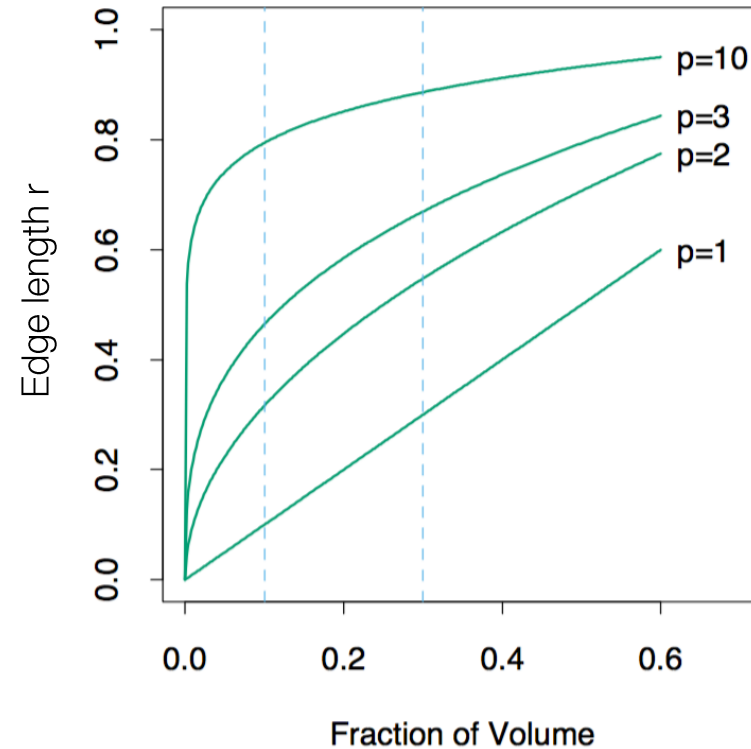
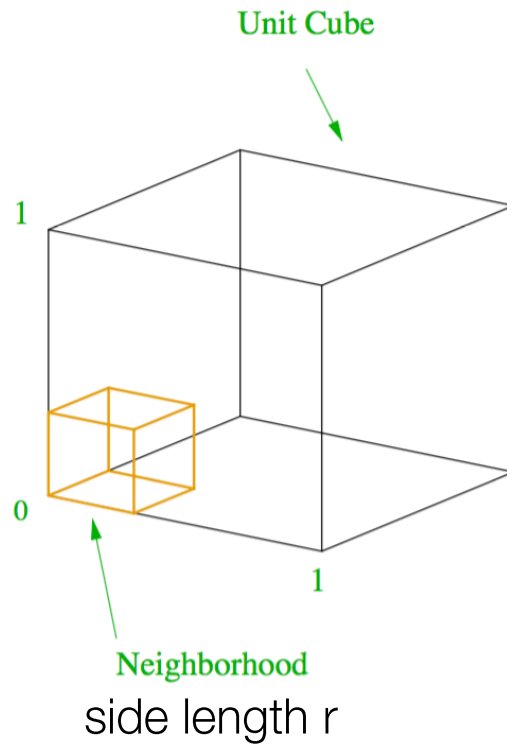


1-NN classification: Theoretical guarantees

1-NN classification: Theoretical guarantees

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 $n \rightarrow \infty$ the 1-NN classification rule has error at most twice the Bayes error rate.

Curse of dimensionality, example 1

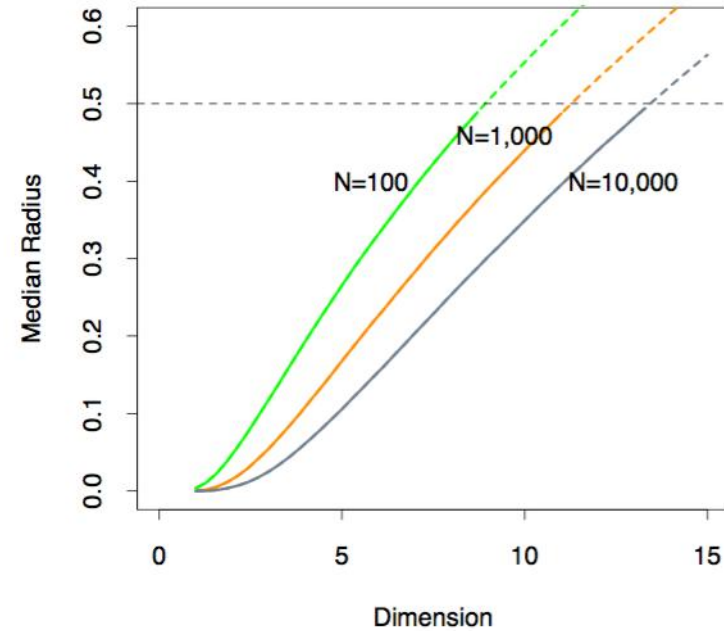
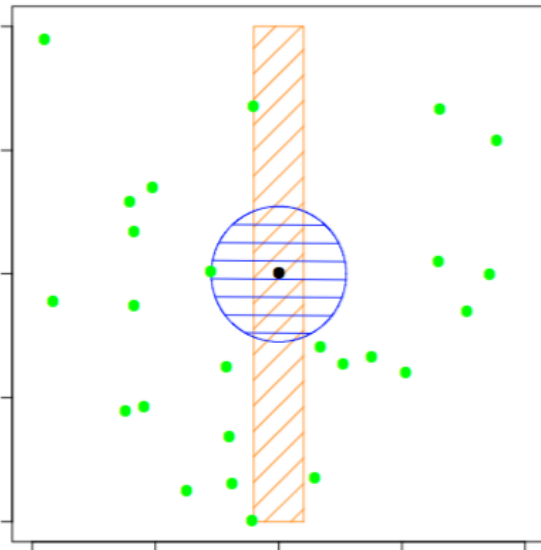


X is uniformly distributed over $[0, 1]^p$. What is $\mathbb{P}(X \in [0, r]^p)$?

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

Curse of dimensionality, example 2

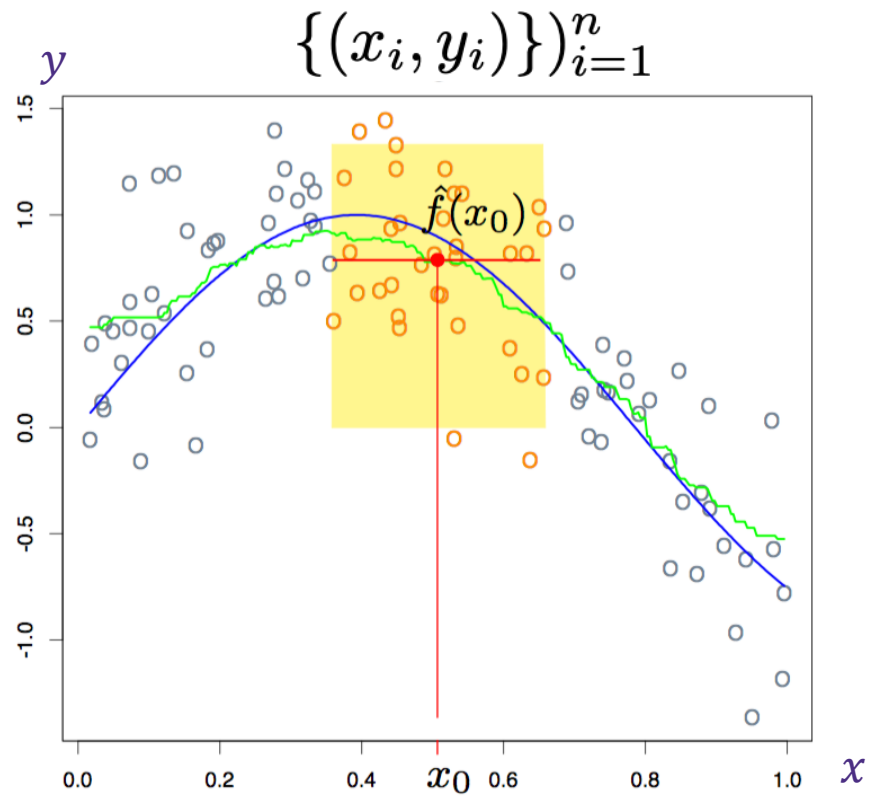
$\{X_i\}_{i=1}^n$ are uniformly distributed over $[-.5, .5]^p$.



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

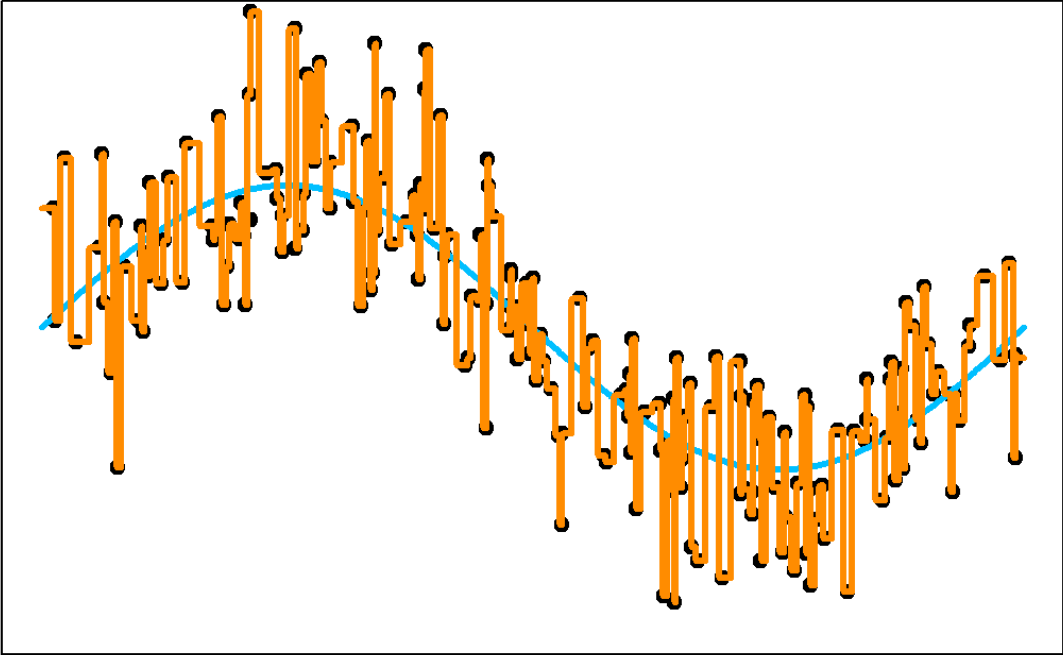
How many samples do we need so that a median Euclidean distance is within r ?

Nearest neighbor regression

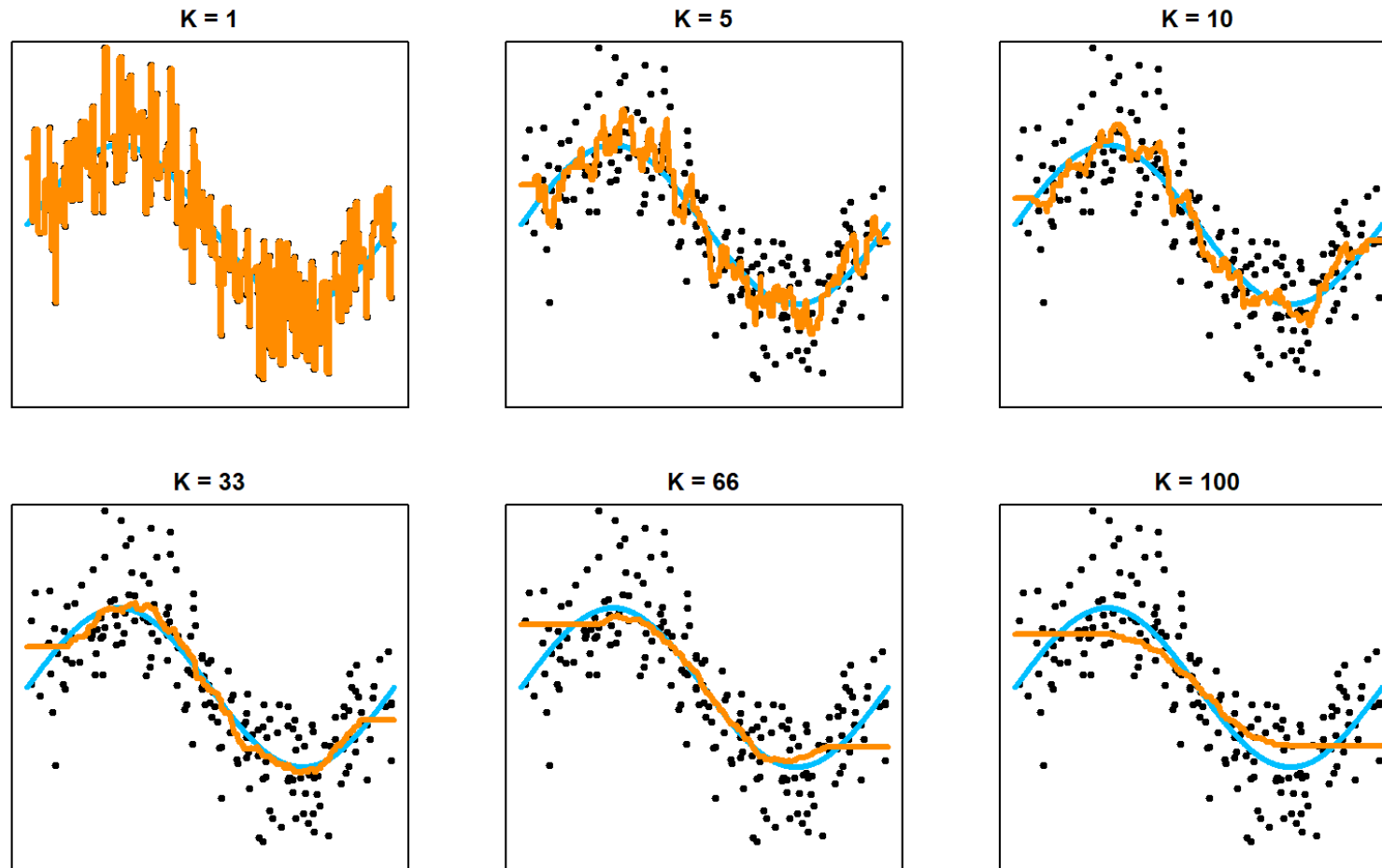


Overfitting

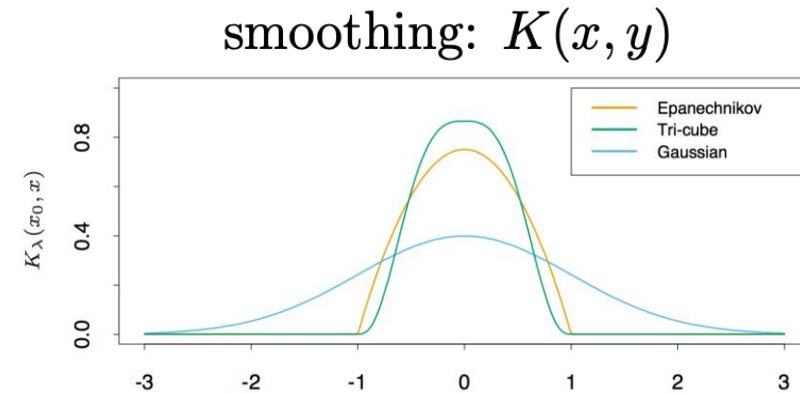
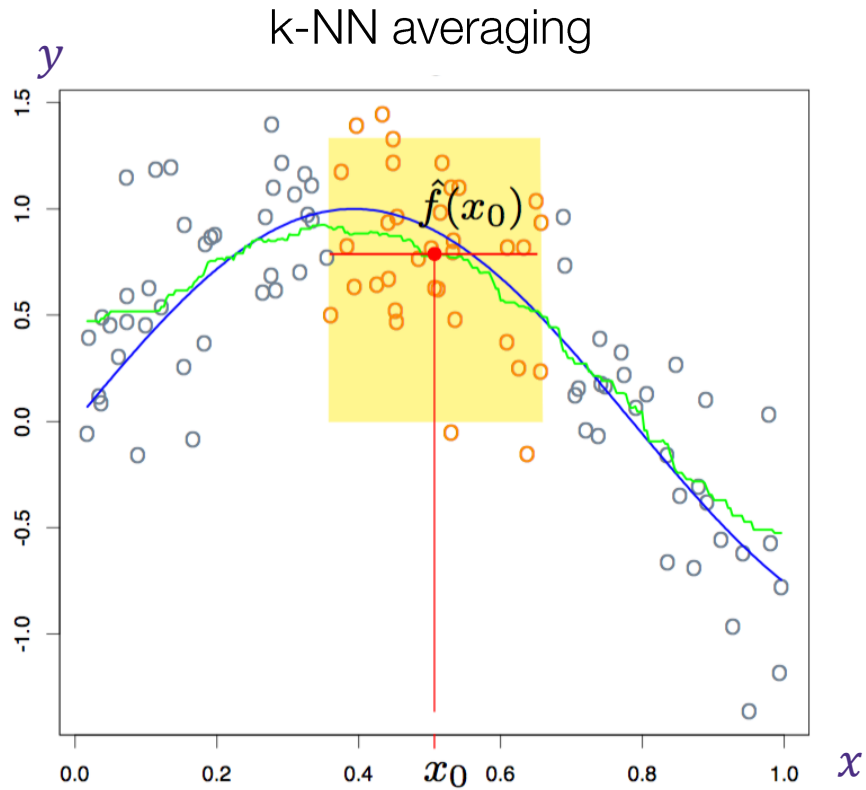
1-Nearest Neighbor Regression



Bias vs variance

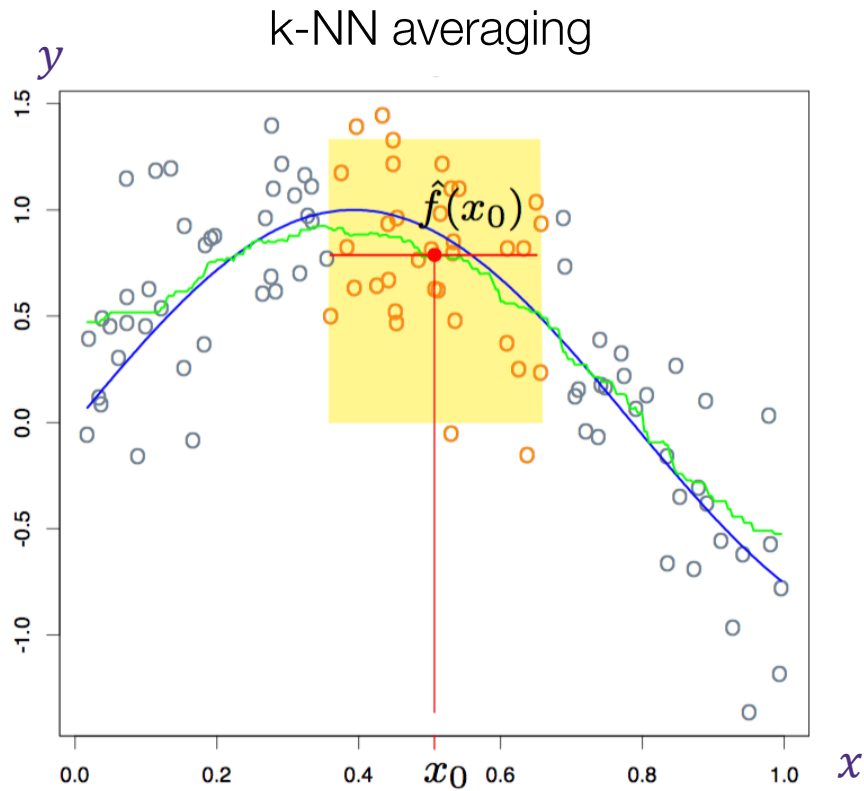


Smoothed nearest neighbor regression

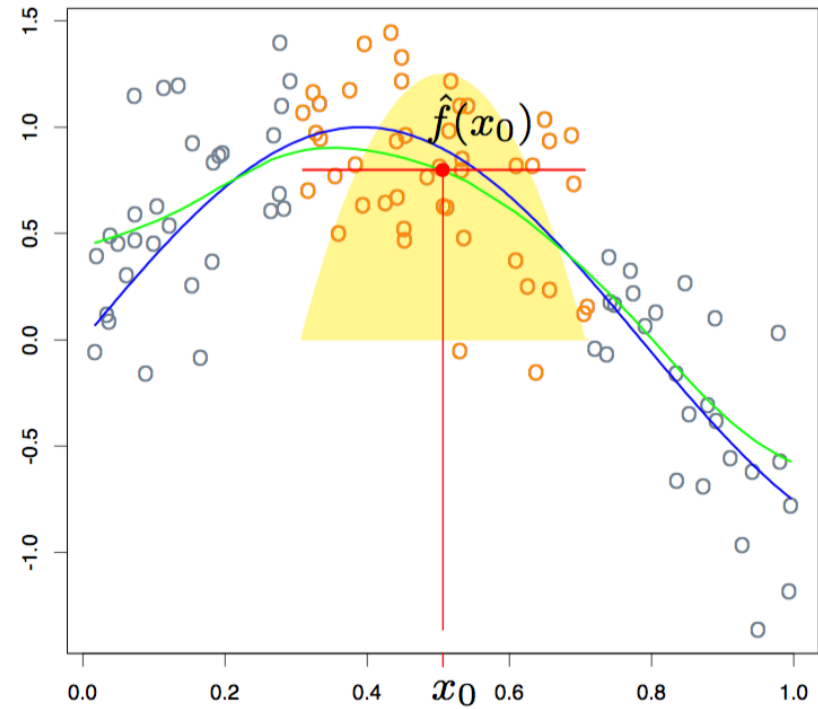


$$\hat{f}(x) = \frac{\sum_{i=1}^n K(x, x^{(i)}) y^{(i)}}{\sum_{i=1}^n K(x, x^{(i)})}$$

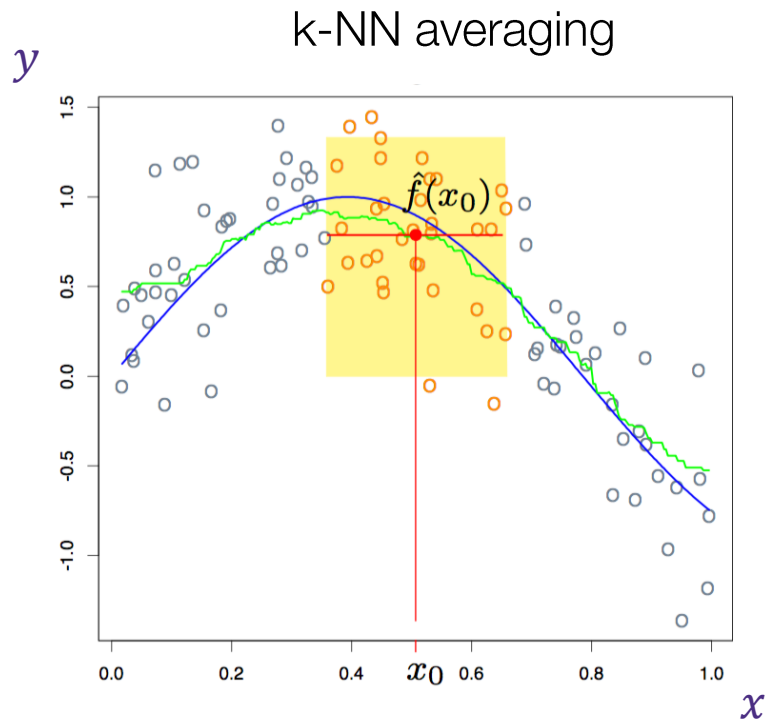
Smoothed nearest neighbor regression



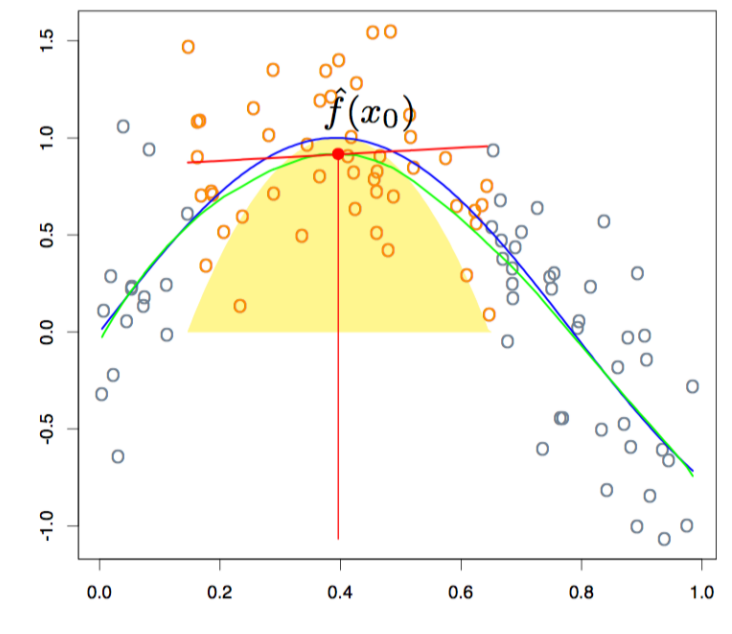
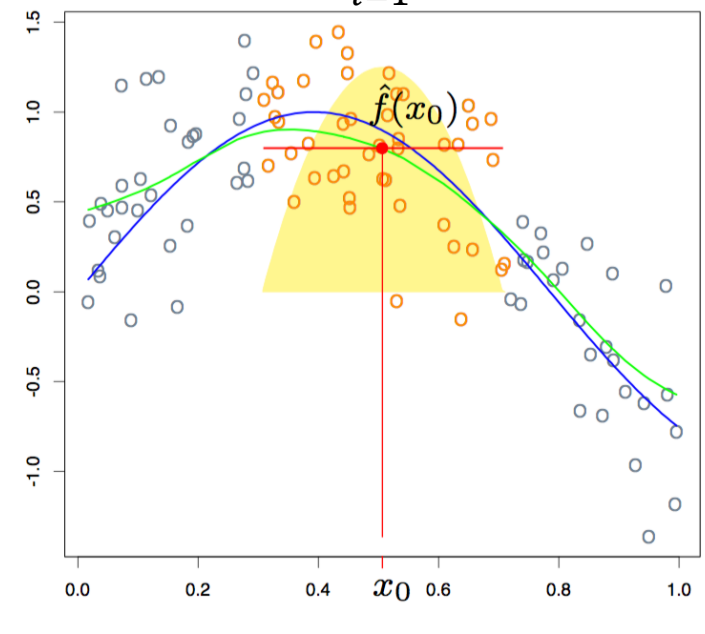
$$\hat{f}(x) = \frac{\sum_{i=1}^n K(x, x^{(i)})y^{(i)}}{\sum_{i=1}^n K(x, x^{(i)})}$$



Locally linear regression



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



Have we seen non-parametric methods before?

- Kernel methods can be non-parametric:

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

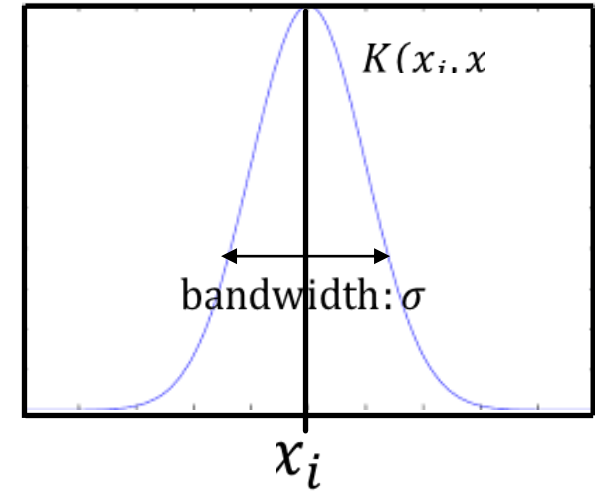
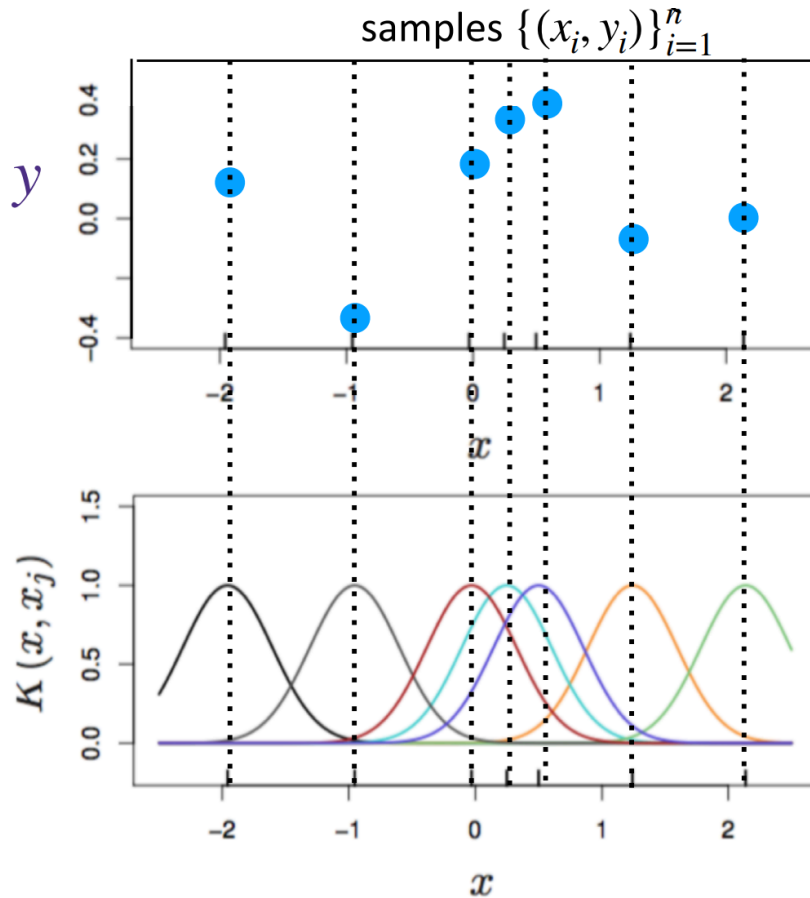
parameters goes up with # data

- Compare with (smoothed) nearest neighbors:

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

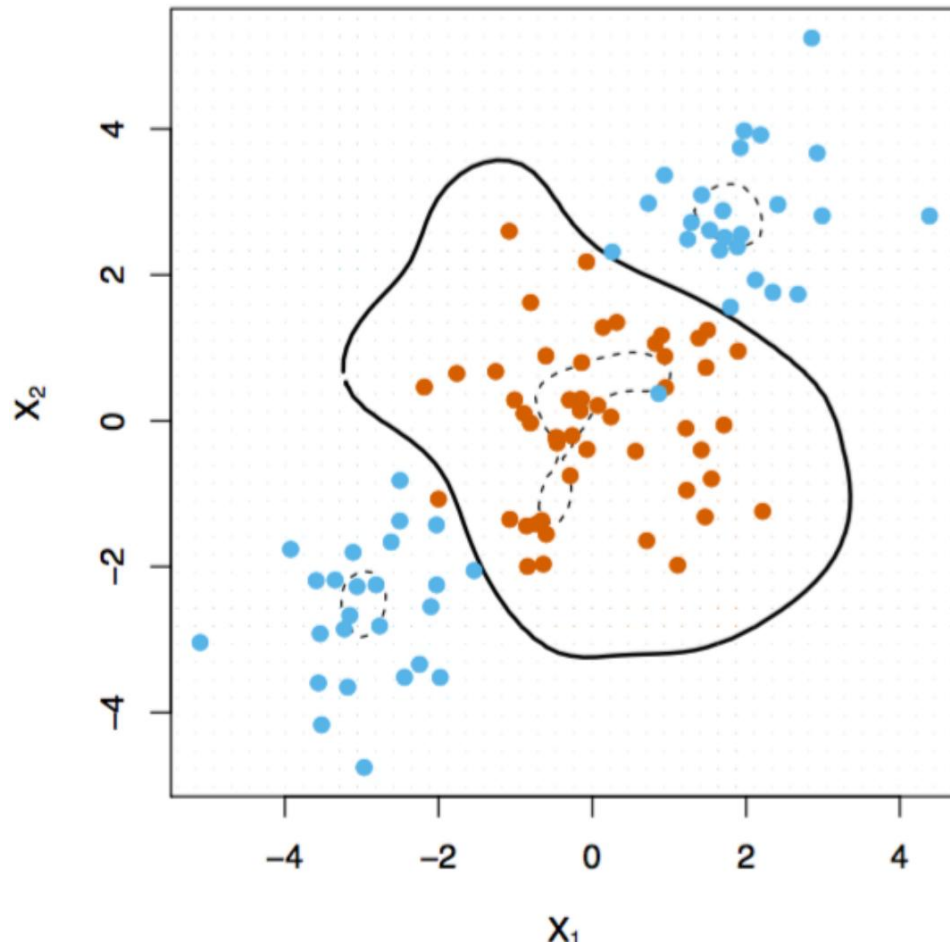
The Radial Basis Function (RBF) kernel $\exp\left(-\frac{\|x^{(i)} - x\|_2^2}{2\sigma^2}\right)$

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

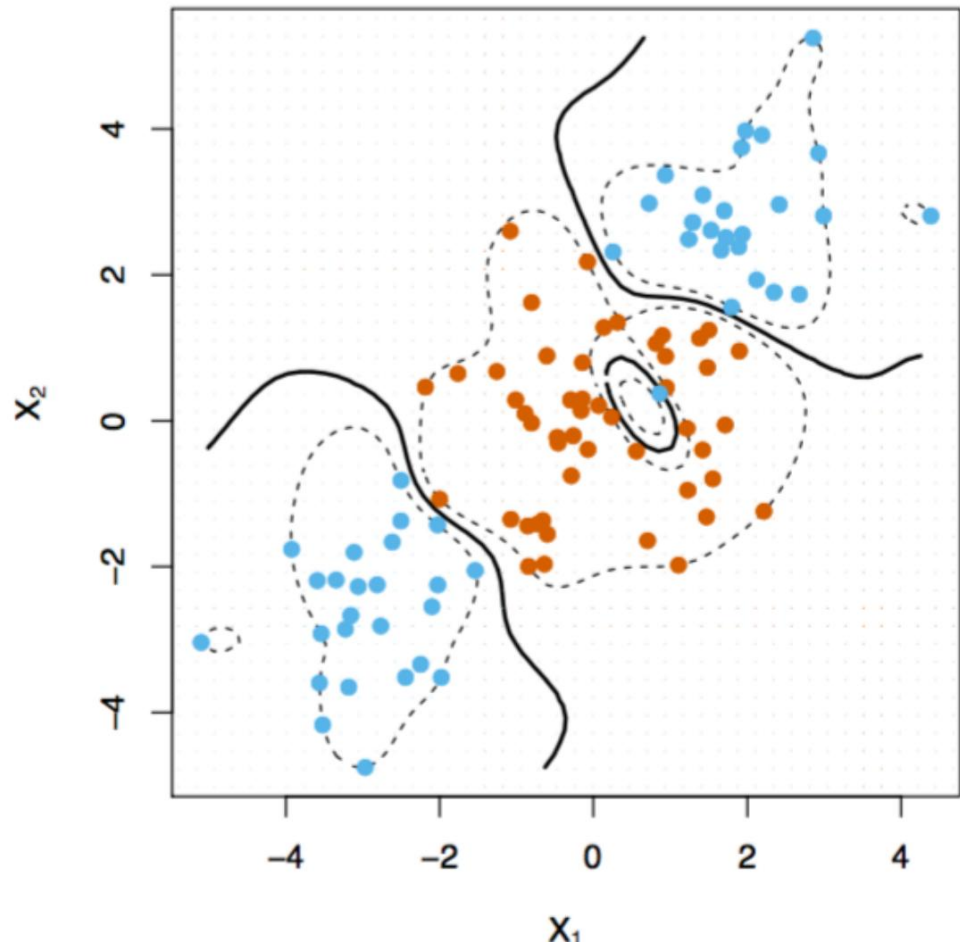


The Radial Basis Function (RBF) kernel $\exp\left(-\frac{\|x^{(i)}-x\|_2^2}{2\sigma^2}\right)$

Bandwidth σ is large enough



Bandwidth σ is small



Kernel methods

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

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

Takeaways

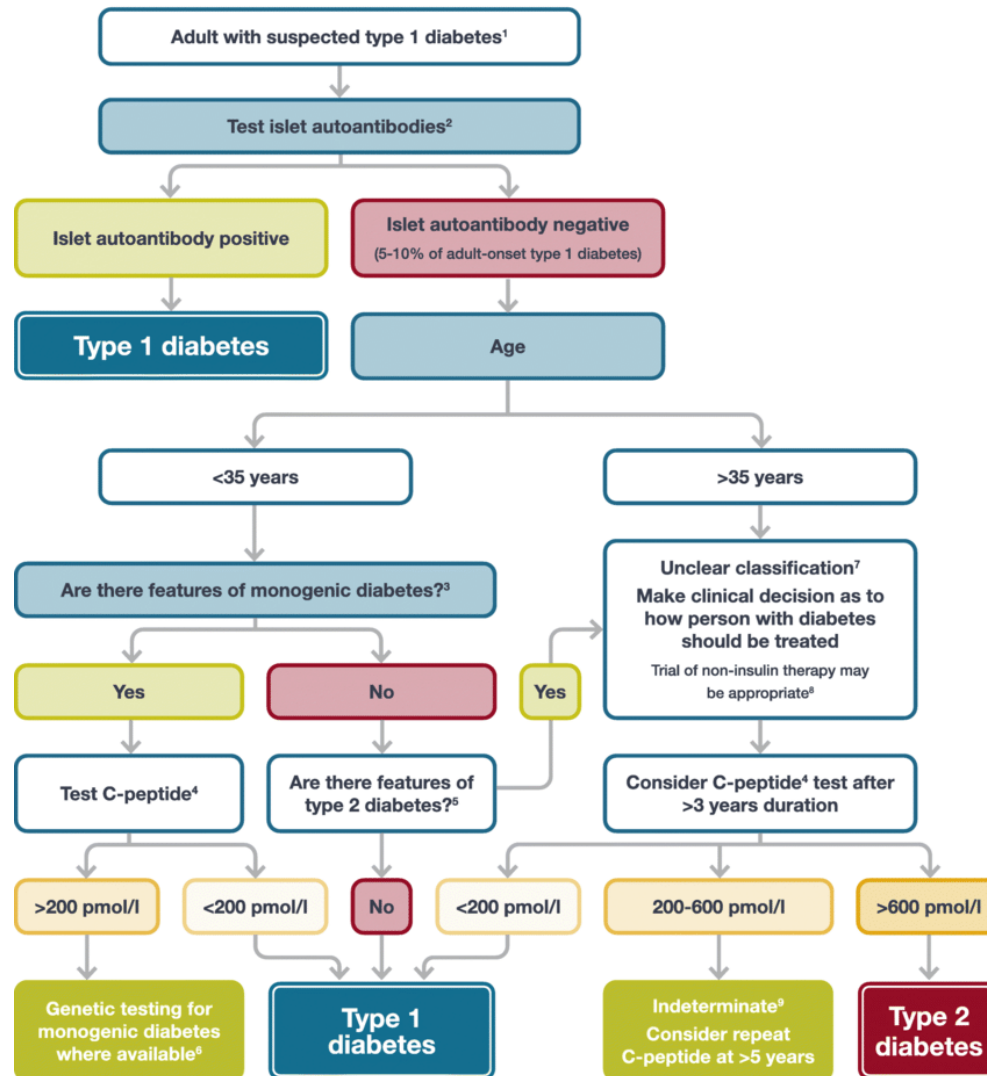
- k-NN is very simple to explain and implement
- No training! But inference can still be computationally demanding.
- You can use other forms of distance (not just Euclidean)
- Smoothing and local linear regression can improve performance (at the cost of higher variance)
- With a lot of data, “local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

Non-parametric methods: Trees

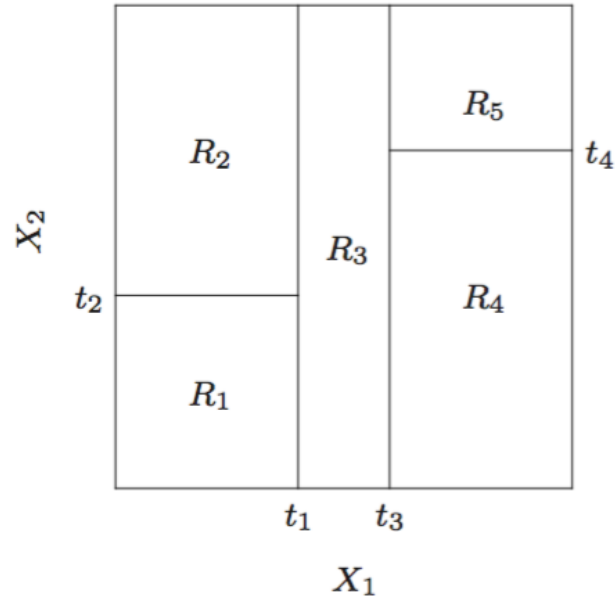
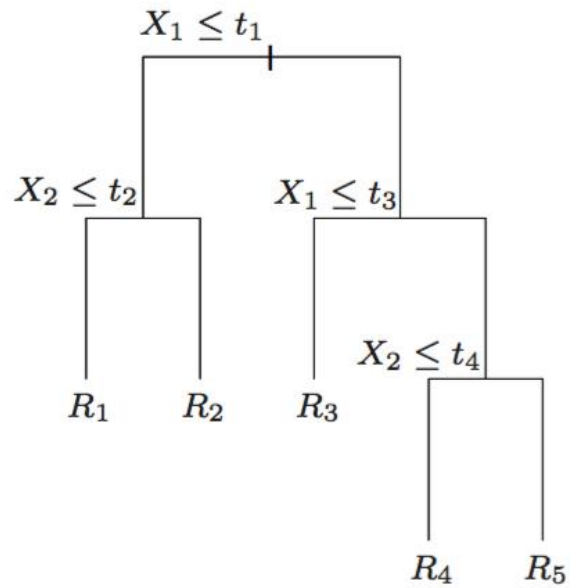
CSE 446/546

Sewoong Oh & Pang Wei Koh

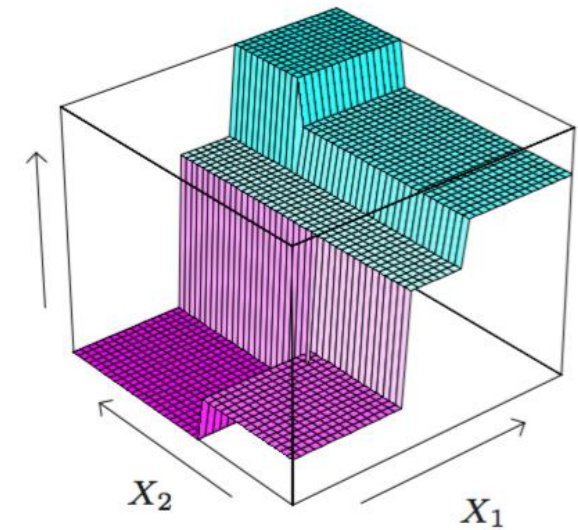
Flow chart for investigation of suspected type 1 diabetes in newly diagnosed adults, based on data from White European populations



Decision / Regression trees



$$f(x) = \sum_{m=1}^M c_m I(x \in R_m).$$



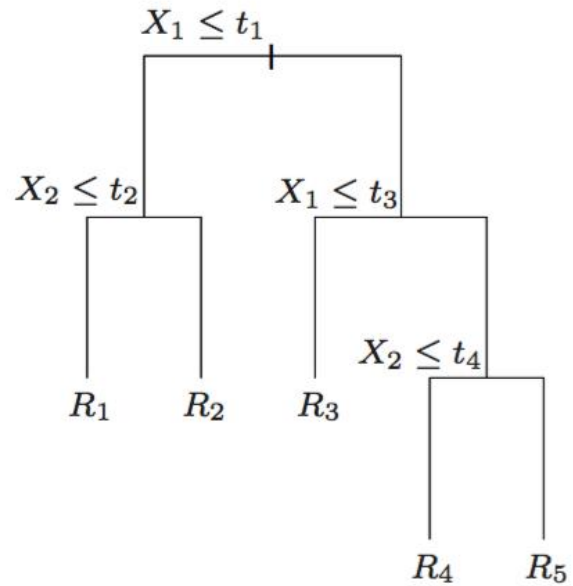
Generic algorithm for building trees

1. Start from empty decision tree
2. Recursively, for each node:
 - Iterate through all features and compute how good it'd be to split on each feature
 - Split on the “best” feature
3. Prune

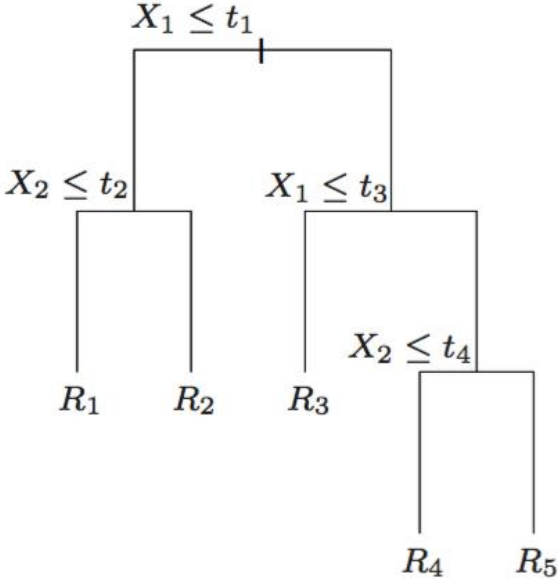
Design choices:

- Termination condition
- Tree complexity
- Splitting criterion
- Pruning

Splitting regression trees



Splitting decision trees



Interpreting trees

Trees are “easy” to interpret:

- You can explain how the classifier came to the conclusion it did

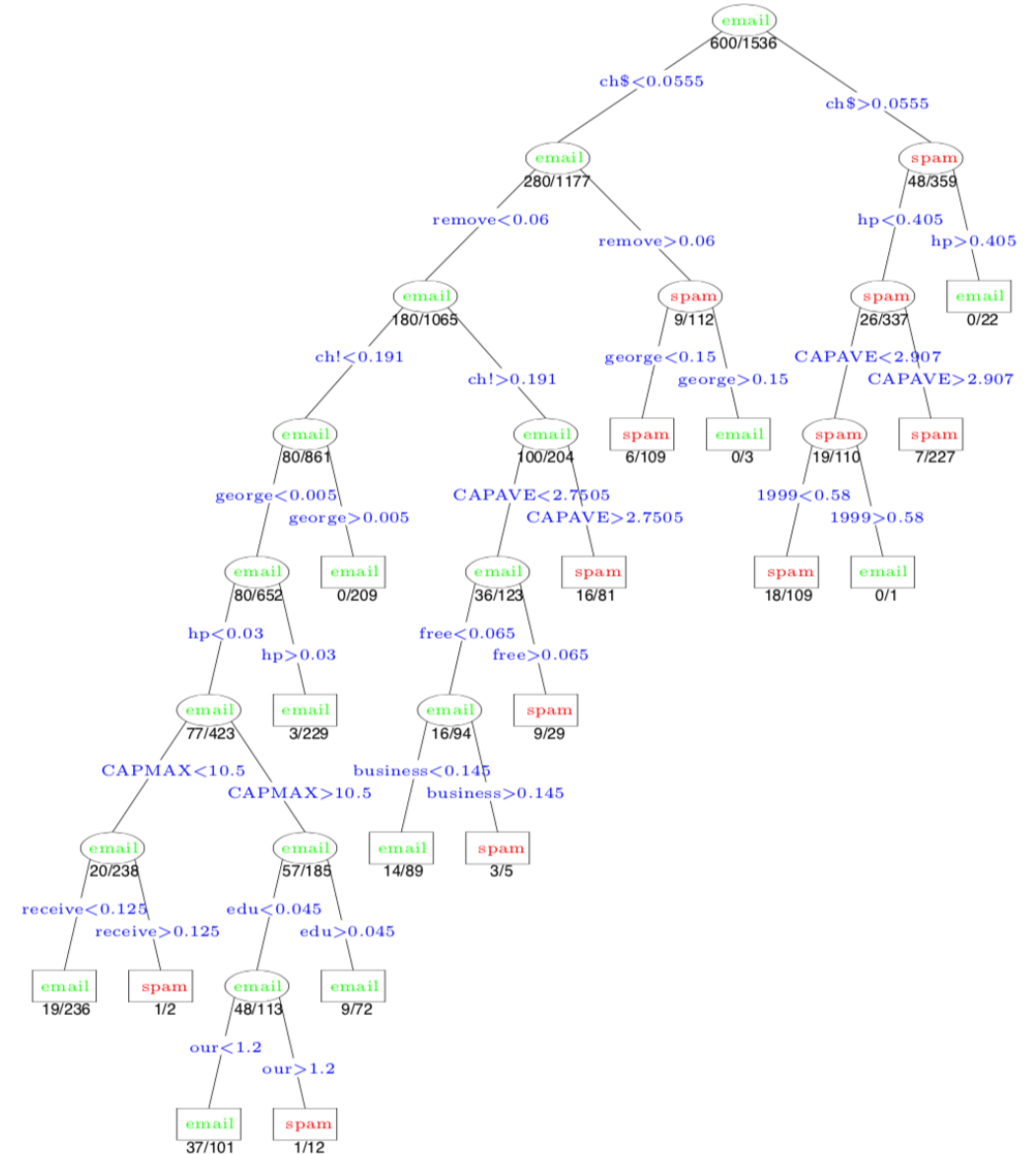
Interpreting trees

Trees are “easy” to interpret:

- You can explain how the classifier came to the conclusion it did

Trees are hard to interpret:

- Tough to explain why the classifier came to the conclusion it did
- Small changes in data can result in large difference in trees

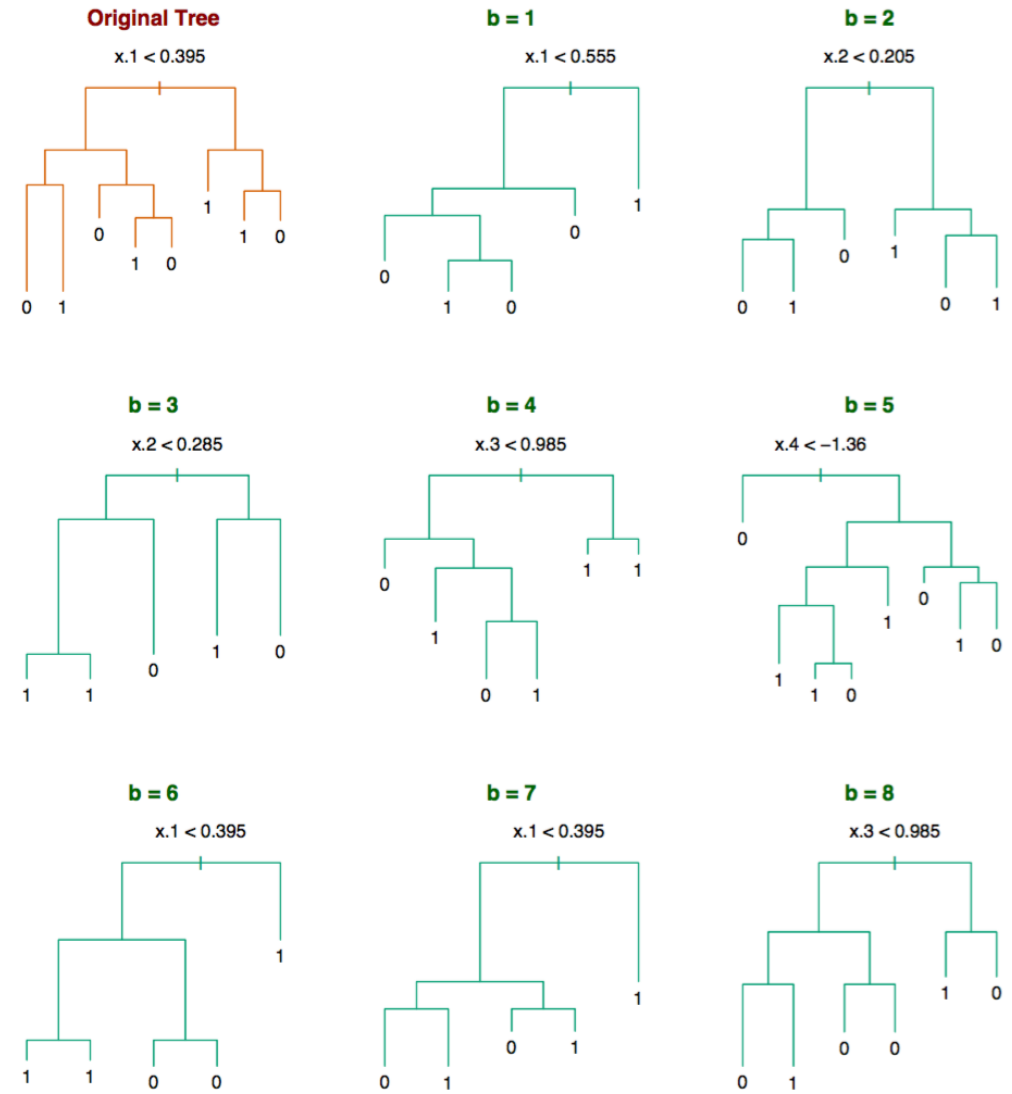


Summary so far

- Trees have bias, variance
- Deal with categorical variables well
- Intuitive, “interpretable”
- Good software exists
- Some theoretical guarantees

Random forests

- Forest = many trees
- Tree methods have low bias but high variance
- We can reduce variance by constructing many “lightly correlated” trees and averaging them
- Bagging: Bootstrap aggregating



Random forests

Algorithm 15.1 *Random Forest for Regression or Classification.*

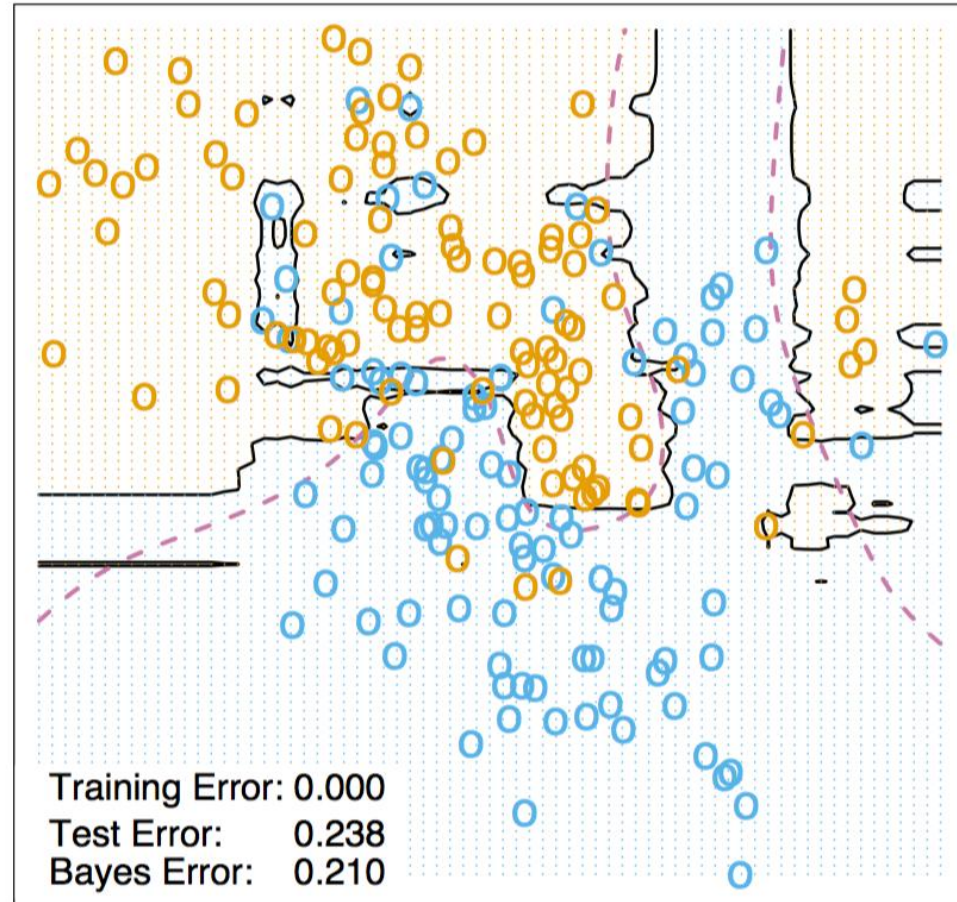
1. For $b = 1$ to B :
 - (a) Draw a bootstrap sample \mathbf{Z}^* of size N from the training data.
 - (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
 - i. Select m variables at random from the p variables.
 - ii. Pick the best variable/split-point among the m .
 - iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x :

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the b th random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B$.

Random forests: Decision boundary example

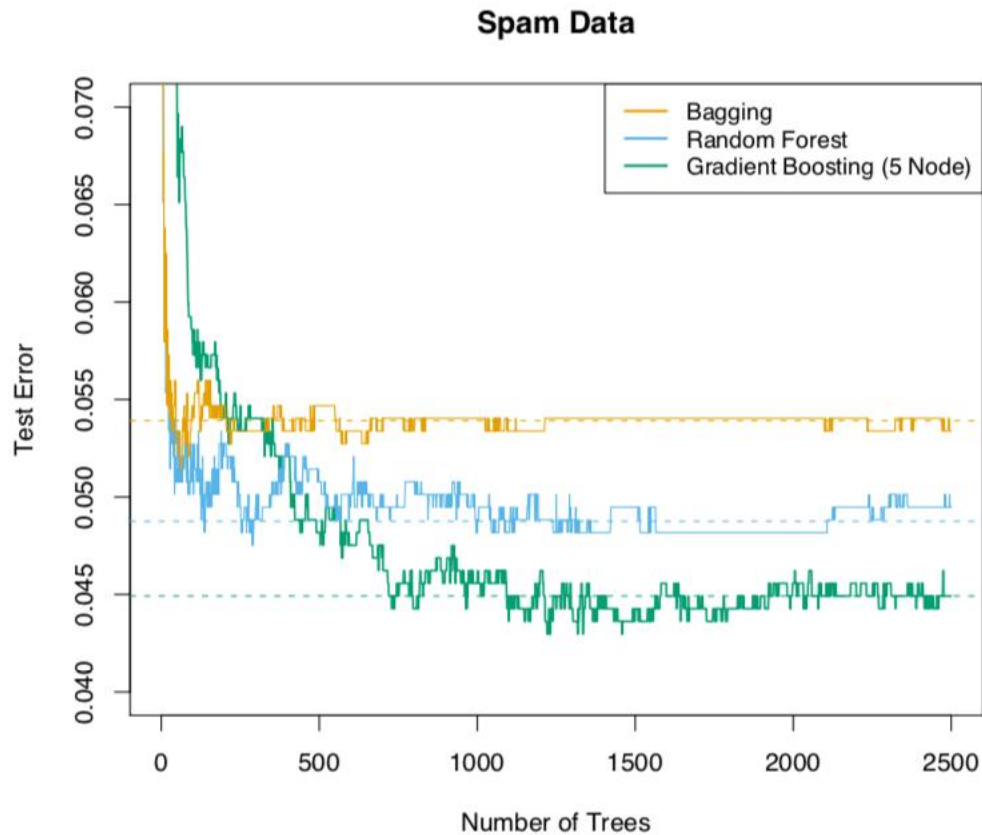


Random forests

Given random variables Y_1, Y_2, \dots, Y_B with
 $\mathbb{E}[Y_i] = y$, $\mathbb{E}[(Y_i - y)^2] = \sigma^2$, $\mathbb{E}[(Y_i - y)(Y_j - y)] = \rho\sigma^2$

$$\mathbb{E}\left[\left(\frac{1}{B} \sum_{i=1}^B Y_i - y\right)^2\right] =$$

The power of weakly correlated predictors



Bagging: Averaged trees on bootstrapped datasets using all d features

Random forest: Averaged trees on bootstrapped datasets using m randomly selected features

Takeaway: reducing correlation improves performance!

Summary so far

- Random forests have bias, variance
- Deal with categorical variables well
- Not that intuitive nor “interpretable”
- Gives some notion of confidence estimates
- Good software exists
- Some theoretical guarantees

Boosting and additive models

Instead of ensembling bootstrapped models, can we:

- Keep the idea of ensembling / combining simpler models, but
- Not necessarily have the models be identically distributed?

Key idea: Given a current collection of models, add a new model that focuses on what the previous models got wrong

Additive models

Forward stagewise additive models

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M :
 - (a) Compute

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$

- (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.
-

Forward stagewise additive models

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M :
 - (a) Compute

$$(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$

- (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.
-

Gradient boosting

A brief history of boosting

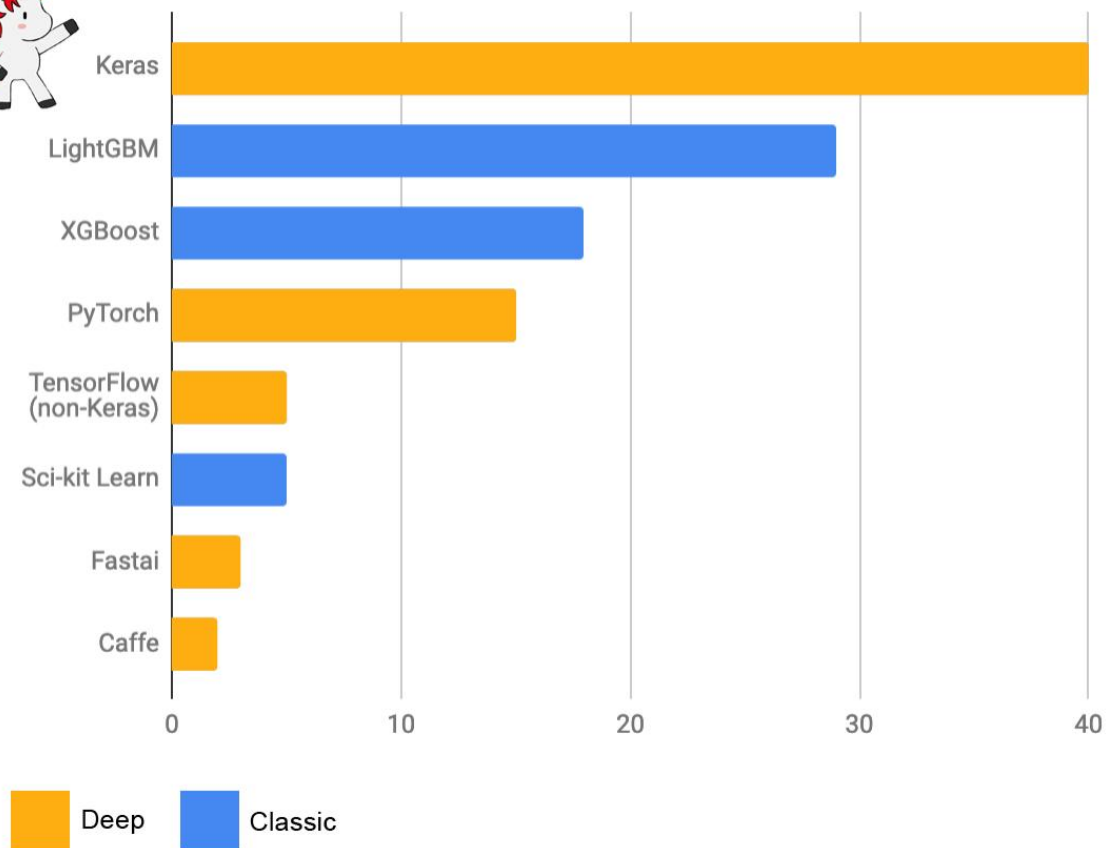
- 1988 Kearns and Valiant: “Can weak learners be combined to create a strong learner?”
- 1990 Schapire: “Yup, in theory”
- 1995 Schapire and Freund: “Practical for 0/1 loss” -> AdaBoost
- 2001 Friedman: “Practical for arbitrary losses”
- 2014 Tianqi Chen: “Scale it up!” -> XGBoost

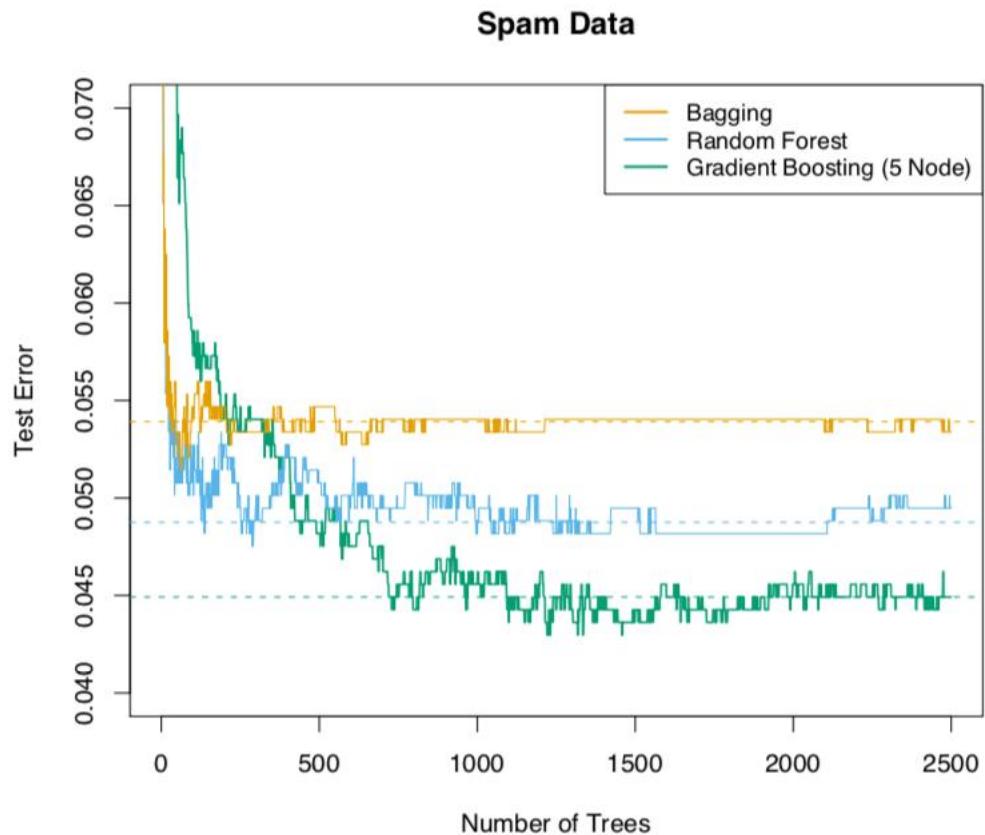


François Chollet ✓ @fchollet · Apr 3, 2019

What machine learning tools do Kaggle champions use? We ran a survey among teams that ranked in the *top 5* of a competition since 2016.

Primary ML software tool used by top-5 teams on Kaggle in each competition (n=120)





Bagging: Averaged trees on bootstrapped datasets using all d features

Random forest: Averaged trees on bootstrapped datasets using m randomly selected features

Boosting: Learned combinations of trees

Takeaways

- Single trees: low bias, high variance
- Ensembles: low bias, (relatively) low variance
- Bagging averages many lightly dependent models to reduce variance
 - Random forests: same but with random subset of features
- Boosting learns a linear combination of high bias, highly dependent classifiers to reduce error
- Gradient boosted trees are commonly used for categorical data

Bonus: Non-parametric methods today

- Are neural networks non-parametric?
 - Technically no, but...

Retrieval-augmented generation and search agents

What's the state-of-the-art in immunotherapy for cervical cancer?

... Pembrolizumab can extend progression-free and overall survival in patients with persistent, recurrent, or metastatic PD-L1-positive cervical cancer [1] and locally advanced cervical cancer [2]...

[1] Colombo et al., KEYNOTE-826, 2021

[2] Lorusso et al., KEYNOTE-A18, 2024

Attribution

A diagram with three blue arrows pointing towards the references. One arrow points from the word 'Attribution' to the reference [1]. Another arrow points from the word 'Up-to-date' to the reference [2]. A third arrow points from the word 'Credible sources' to the reference [1].

Up-to-date

Credible sources

Next token prediction

Enter text:

One, two,



3198 11 734 11

Prediction

| # | probs | next token ID | predicted next token |
|---|--------|---------------|----------------------|
| 0 | 39.71% | 1115 | three |
| 1 | 16.97% | 290 | and |
| 2 | 7.55% | 734 | two |
| 3 | 3.76% | 1440 | four |
| 4 | 2.76% | 393 | or |
| 5 | 2.18% | 1936 | five |
| 6 | 1.57% | 530 | one |
| 7 | 1.43% | 345 | you |
| 8 | 1.15% | 257 | a |
| 9 | 0.84% | 3598 | seven |

Next token prediction

Enter text:

One, two, three



3198 11 734 11 1115

Prediction

| # | probs | next token ID | predicted next token |
|---|--------|---------------|----------------------|
| 0 | 54.42% | 11 | , |
| 1 | 5.45% | 1399 | ... |
| 2 | 4.82% | 13 | . |
| 3 | 4.51% | 290 | and |
| 4 | 2.72% | 986 | ... |
| 5 | 2.51% | 25 | : |
| 6 | 1.50% | 393 | or |
| 7 | 1.23% | 3926 | ... |
| 8 | 0.85% | 553 | , |
| 9 | 0.84% | 960 | — |

Next token prediction

Enter text:

One, two, three,



3198 11 734 11 1115 11

Prediction

| # | probs | next token ID | predicted next token |
|---|--------|---------------|----------------------|
| 0 | 46.44% | 1440 | four |
| 1 | 7.48% | 290 | and |
| 2 | 7.31% | 1936 | five |
| 3 | 2.66% | 393 | or |
| 4 | 2.54% | 2237 | six |
| 5 | 2.09% | 1115 | three |
| 6 | 1.86% | 3863 | maybe |
| 7 | 1.62% | 345 | you |
| 8 | 1.23% | 257 | a |
| 9 | 0.92% | 530 | one |

Next token prediction

Enter text:

One, two, three, four



3198 11 734 11 1115 11 1440

Prediction

| # | probs | next token ID | predicted next token |
|---|--------|---------------|----------------------|
| 0 | 50.14% | 11 | , |
| 1 | 6.66% | 13 | . |
| 2 | 5.91% | 1399 | ... |
| 3 | 3.15% | 25 | : |
| 4 | 2.63% | 290 | and |
| 5 | 2.58% | 986 | ... |
| 6 | 1.42% | 3926 | ... |
| 7 | 1.17% | 553 | , |
| 8 | 1.09% | 960 | — |
| 9 | 1.08% | 526 | ." |

Conditioning on retrieved context

Retrieving the right documents

Takeaways

- Retrieval-augmented language models are nonparametric models
- Compare and contrast with k-NN
- Many design choices
 - How to find documents?
 - How to incorporate retrieved documents into generation?
 - Multimodal retrieval?
- Highly active research area