Announcements

Coronavirus – COVID-19

- Take care of yourself and others around you
- Follow CMU and government guidelines
- We're "here" to help in any capacity that we can
- Use tools like zoom to communicate with each other too!

Zoom

- Let us know if you have issues
- Etiquette: Turn on video when talking or your turn in OH

Feedback: See Piazza post

Announcements

Assignments

- HW6 (written + programming)
  - Due Thu 3/26, 11:59 pm

HW7 (online) next Tue

"Participation" Points

- Polls open until 10 am (EDT) day after lecture
- "Calamity" option announced in recorded lecture
  - Don't select this calamity option or you'll lose credit for one poll (-1) rather than gaining credit for one poll (+1).
- Participation percent calculated as usual

Introduction to Machine Learning

→ Cross-Validation

→ Nonparametric Regression

Instructor: Pat Virtue

Validation

Why do we need validation?

- Choose hyperparameters
- Choose technique
- Help make any choices beyond our parameters 

But now, we have another choice to make!

- How do we split training and validation?

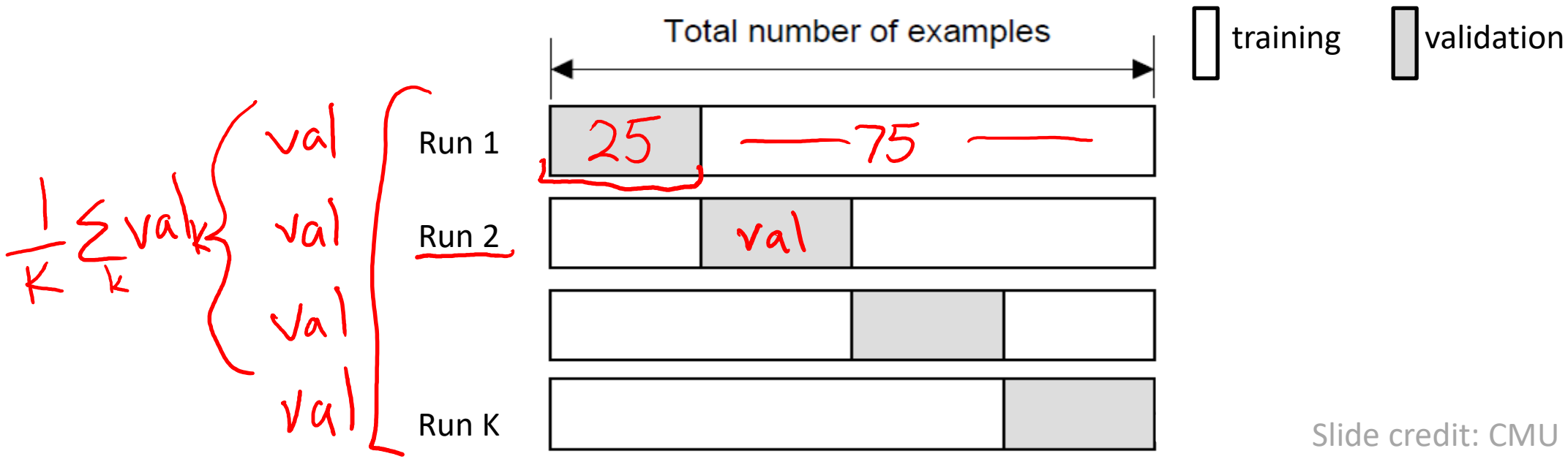
Trade-offs

- More held-out data, better meaning behind validation numbers
- More held-out data, less data to train on!

Cross-validation

K-fold cross-validation

Create K-fold partition of the dataset.
Do K runs: train using K-1 partitions and calculate validation error on remaining partition (rotating validation partition on each run).
Report average validation error

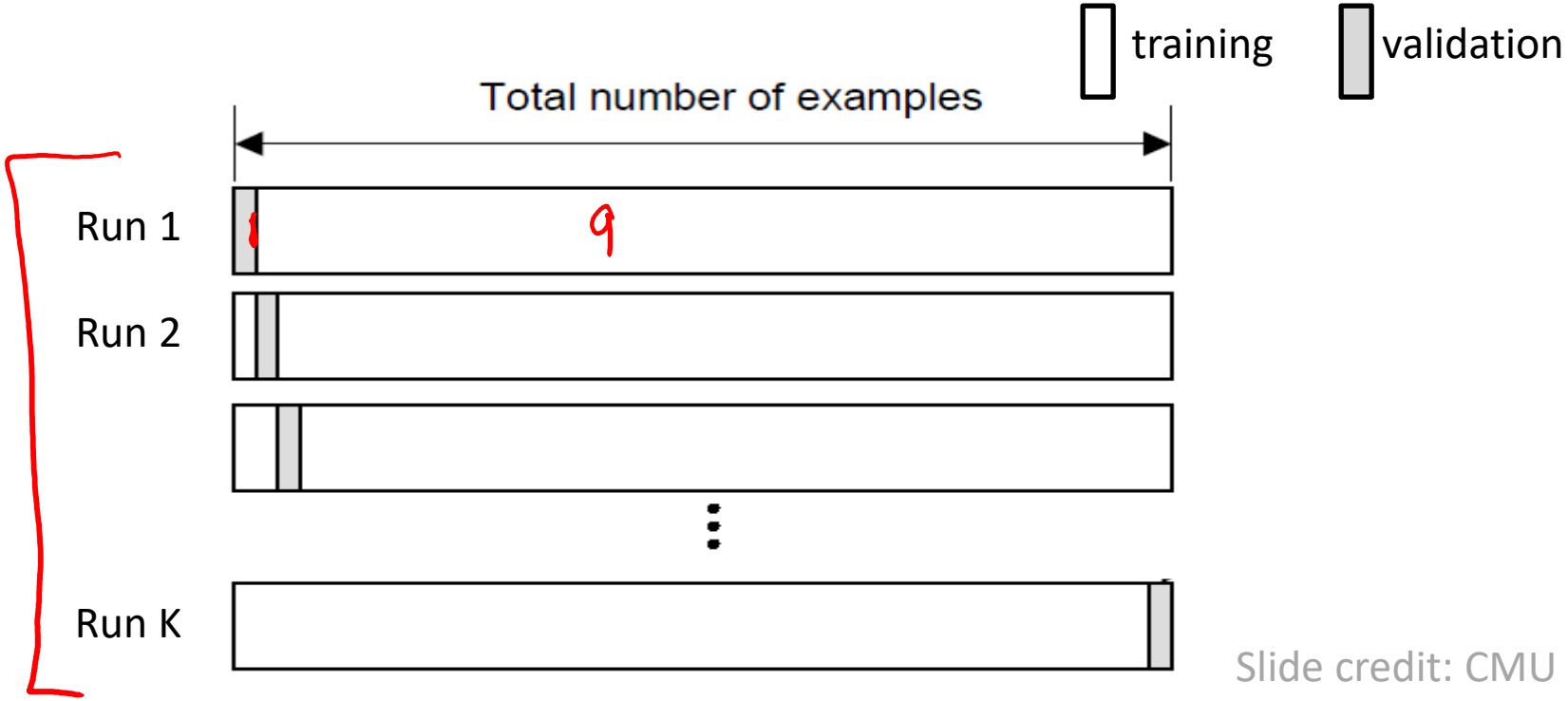


Slide credit: CMU MLD Aarti Singh

Cross-validation

Leave-one-out (LOO) cross-validation

Special case of K-fold with K=N partitions
Equivalently, train on N-1 samples and validate on only one sample per run for N runs



Slide credit: CMU MLD Aarti Singh

Cross-validation

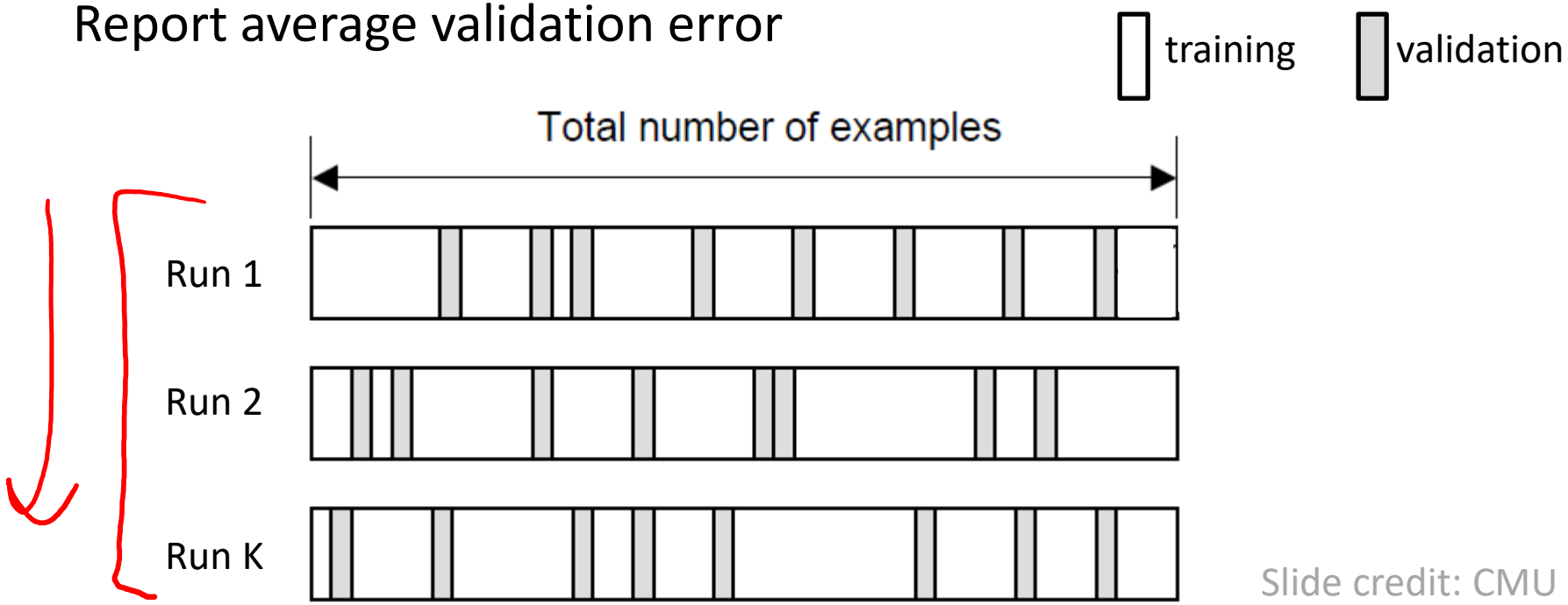
Random subsampling

Randomly subsample a fixed fraction αN ($0 < \alpha < 1$) of the dataset for validation.

Compute validation error with remaining data as training data.

Repeat K times

Report average validation error



Slide credit: CMU MLD Aarti Singh

Practical Issues in Cross-validation

How to decide the values for K and α ?

- Large K
  - + Validation error can approximate test error well
  - - Observed validation error will be unstable (few validation pts)
  - - The computational time will be very large as well (many experiments)

small α

- Small K
  - + The # experiments and, therefore, computation time are reduced
  - + Observed validation error will be stable (many validation pts)
  - - Validation error cannot approximate test error well

bigger α

Common choice: K = 10, α = 0.1 ☺

Slide credit: CMU MLD Aarti Singh

Piazza Poll 1

Say you are choosing amongst 10 values of lambda, and you want to do K=10 fold cross-validation.

How many times do I have to train my model?

A. 0 ← Calamity for λ in [0.1 ... 100] 10

B. 1

C. 10 for k in K=1...10 10

D. 20

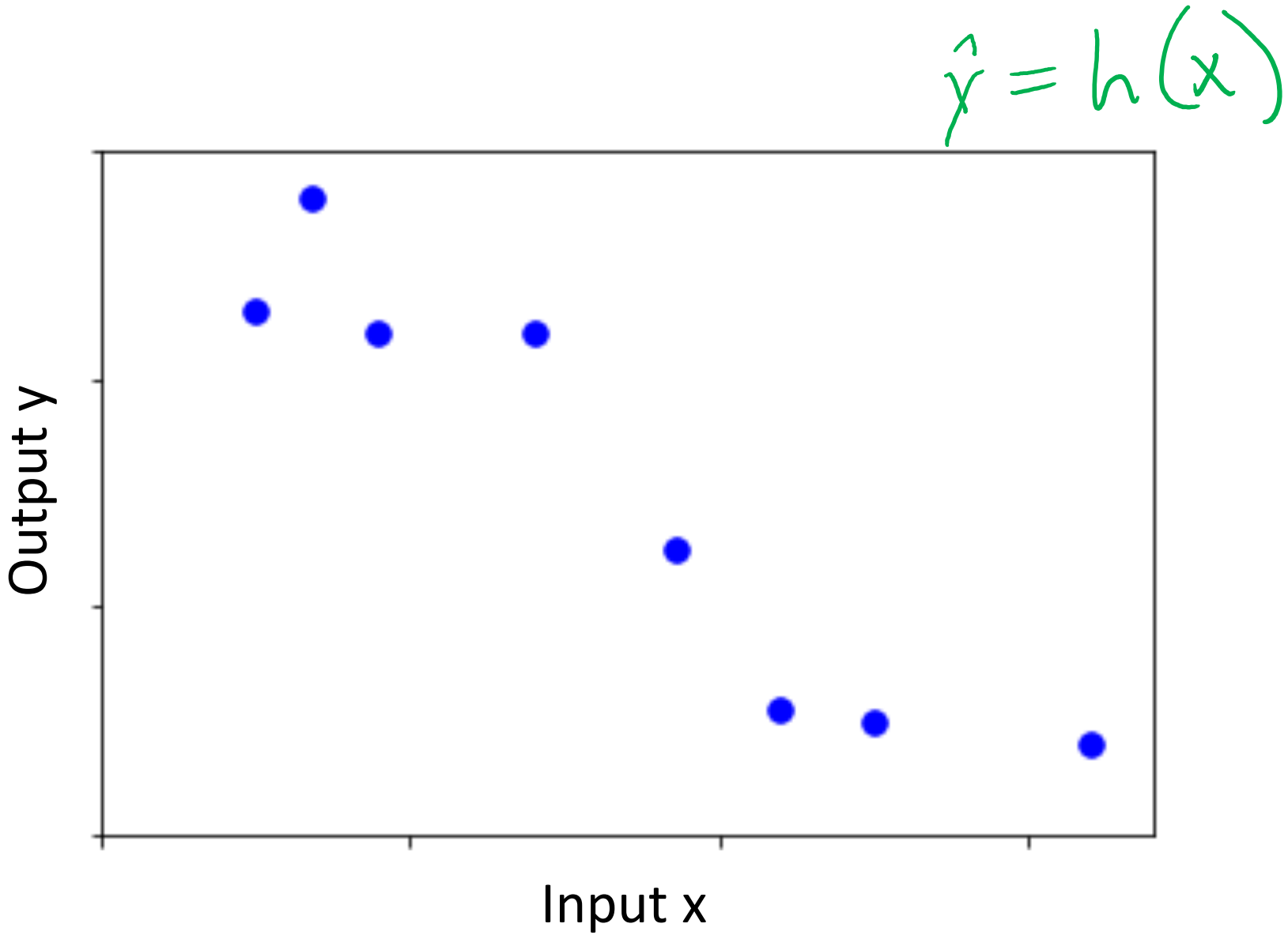
E. 100

F. 10^{10}

→ cv_error[λ]

→

Nonparametric Regression



Reminder: Parametric models

Assume some model (Gaussian, Bernoulli, Multinomial, logistic, network of logistic units, Linear, Quadratic) with fixed number of parameters

- Linear/Logistic Regression, Naïve Bayes, Discriminant Analysis, Neural Networks

Estimate parameters ($\mu, \sigma^2, \theta, w, \beta$) using MLE/MAP and plug in

Pro – need fewer data points to learn parameters

Con – Strong distributional assumptions, not satisfied in practice

Reminder: Nonparametric models

Nonparametric: number of parameters scales with number of training data

- Typically don't make any distributional assumptions
- As we have more data, we should be able to learn more complex models

Example

- Nearest Neighbor (k-Nearest Neighbor) Classifier

Piazza Poll 2

Are decision trees parametric or non-parametric?

~~A.~~ Calamity

B. Parametric

C. Nonparametric

Piazza Poll 2

Are decision trees parametric or non-parametric?

It depends :)

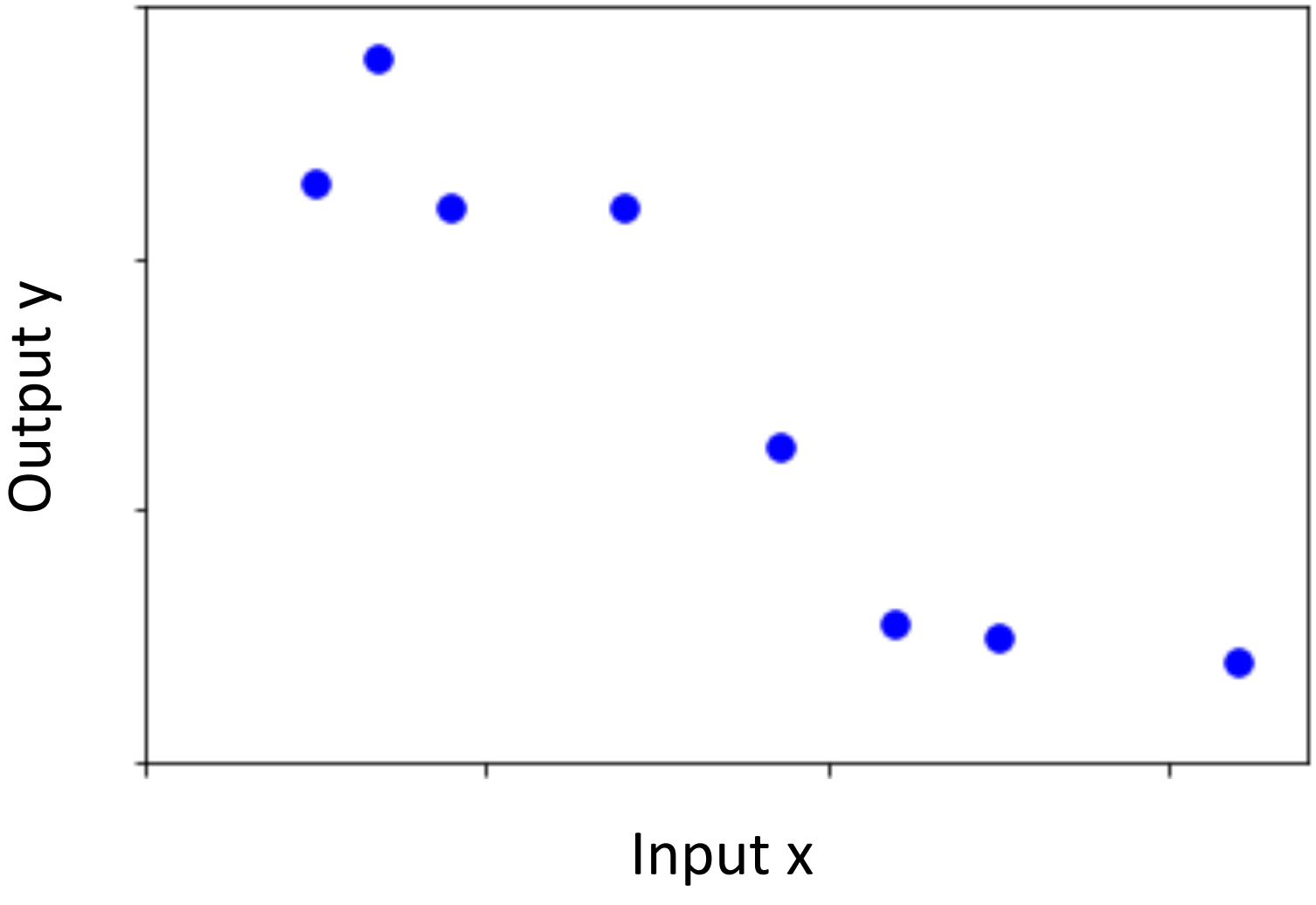
- If no limits on depth or reuse of attributes, then non-parametric
  - Model complexity will grow with data
- If pruned/limited to fix size
  - Parametric
- If attributes only used once
  - Parametric; model complexity is limited by number of features

Trade-offs

- Non-parametric methods have very powerful representation capabilities
- But
  - Easily overfit
  - Can take up memory proportional to training size too

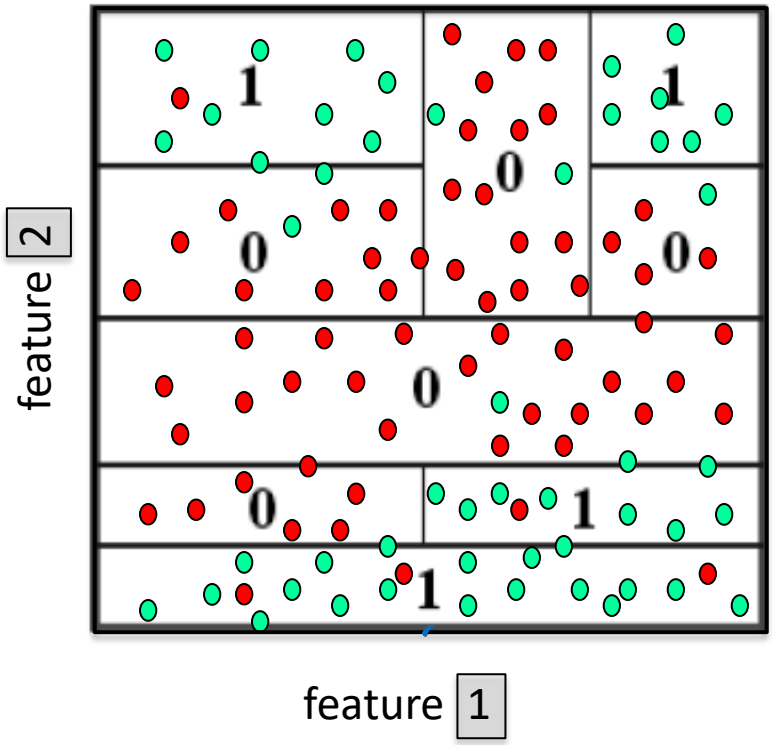
Nonparametric Regression

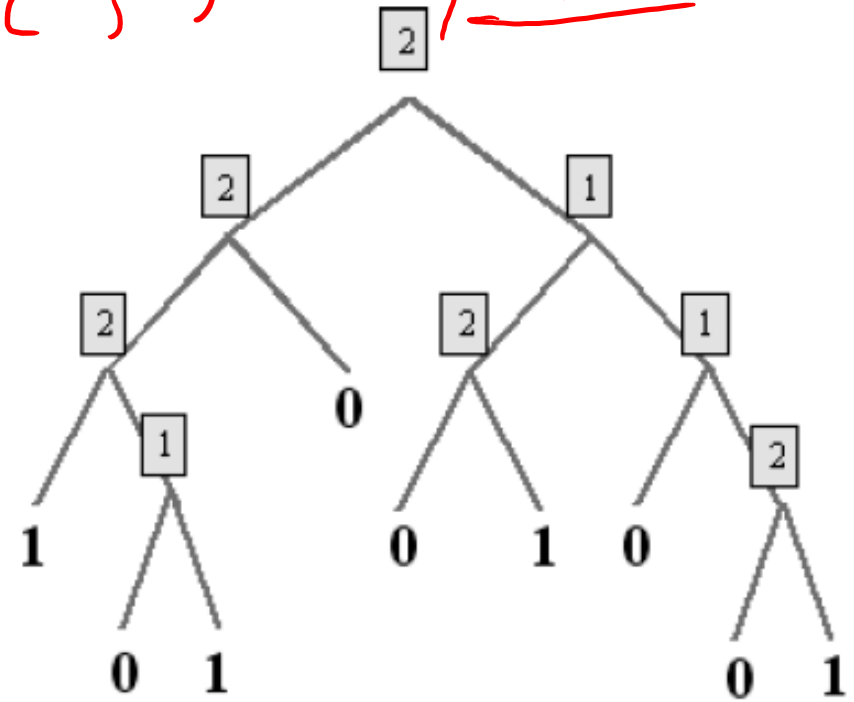
Decision Trees



Dyadic decision trees
(split on mid-points of features)

$x \in \mathbb{R}^2$ $y \in \{0, 1\} \rightarrow y \in \mathbb{R}$

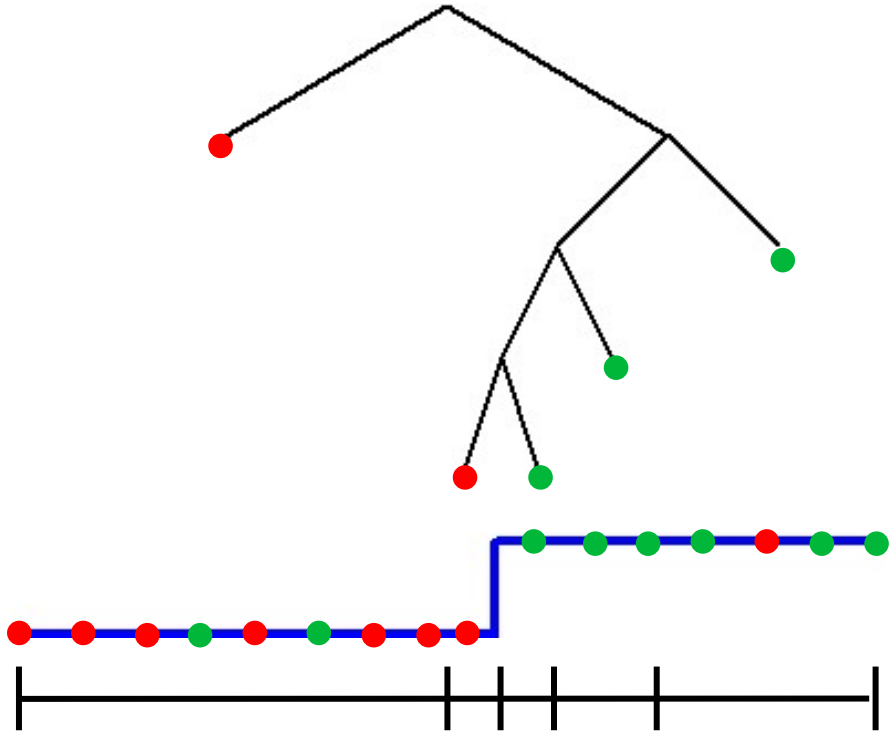




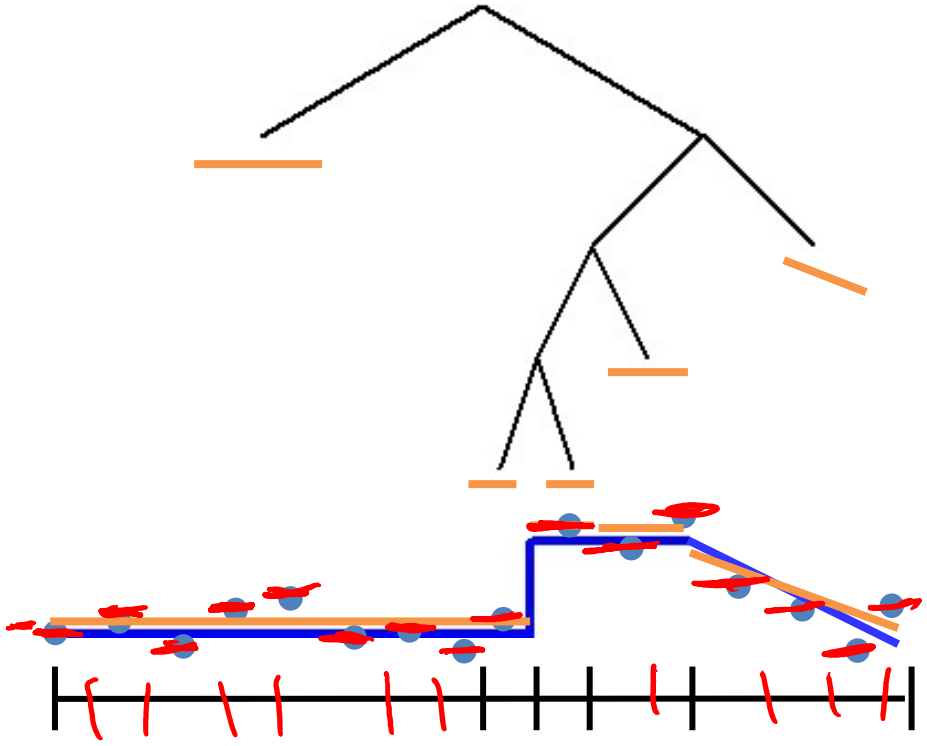
Slide credit: CMU MLD Aarti Singh

How to assign label to each leaf

Classification – Majority vote

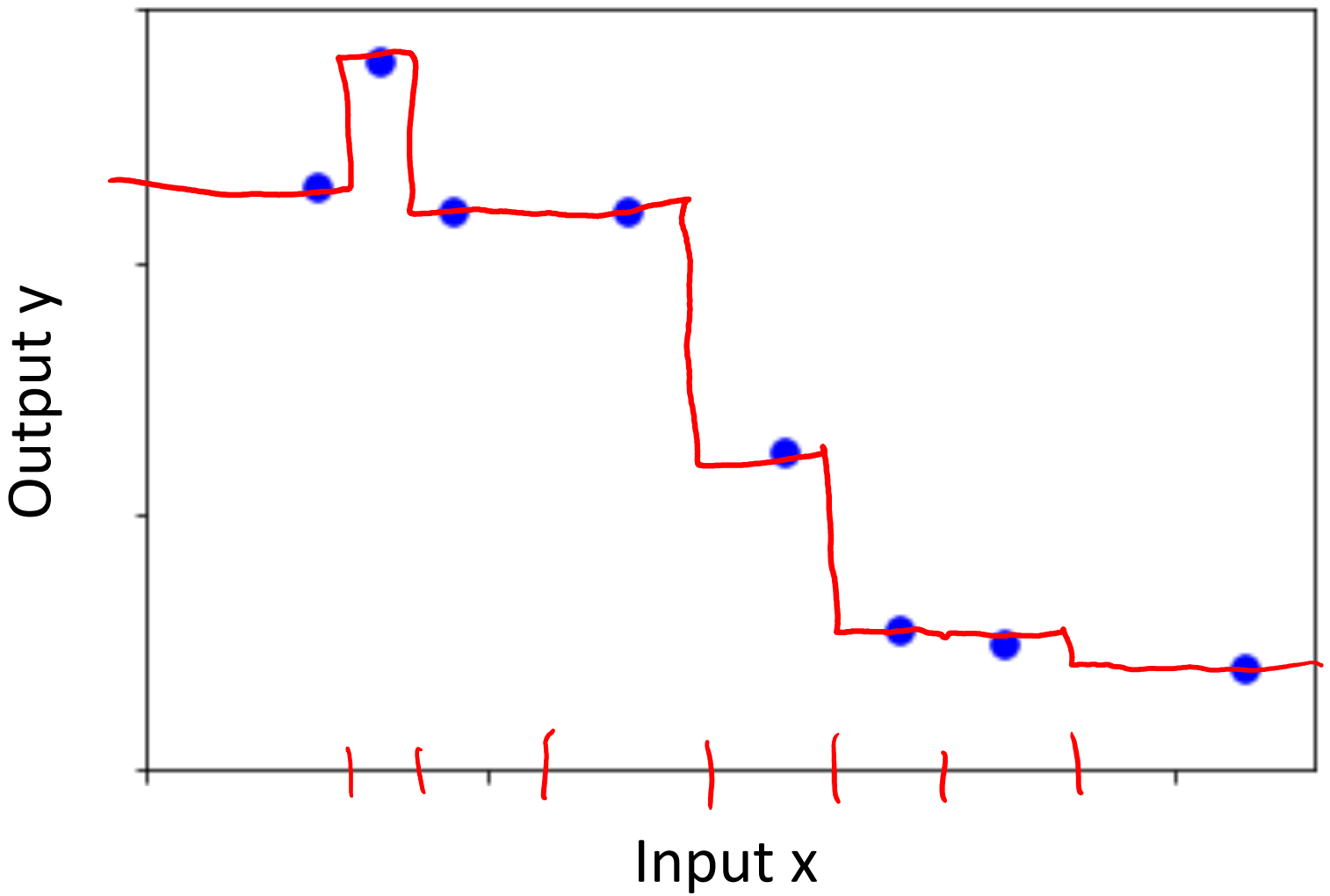


Regression – Constant/Linear/Poly fit



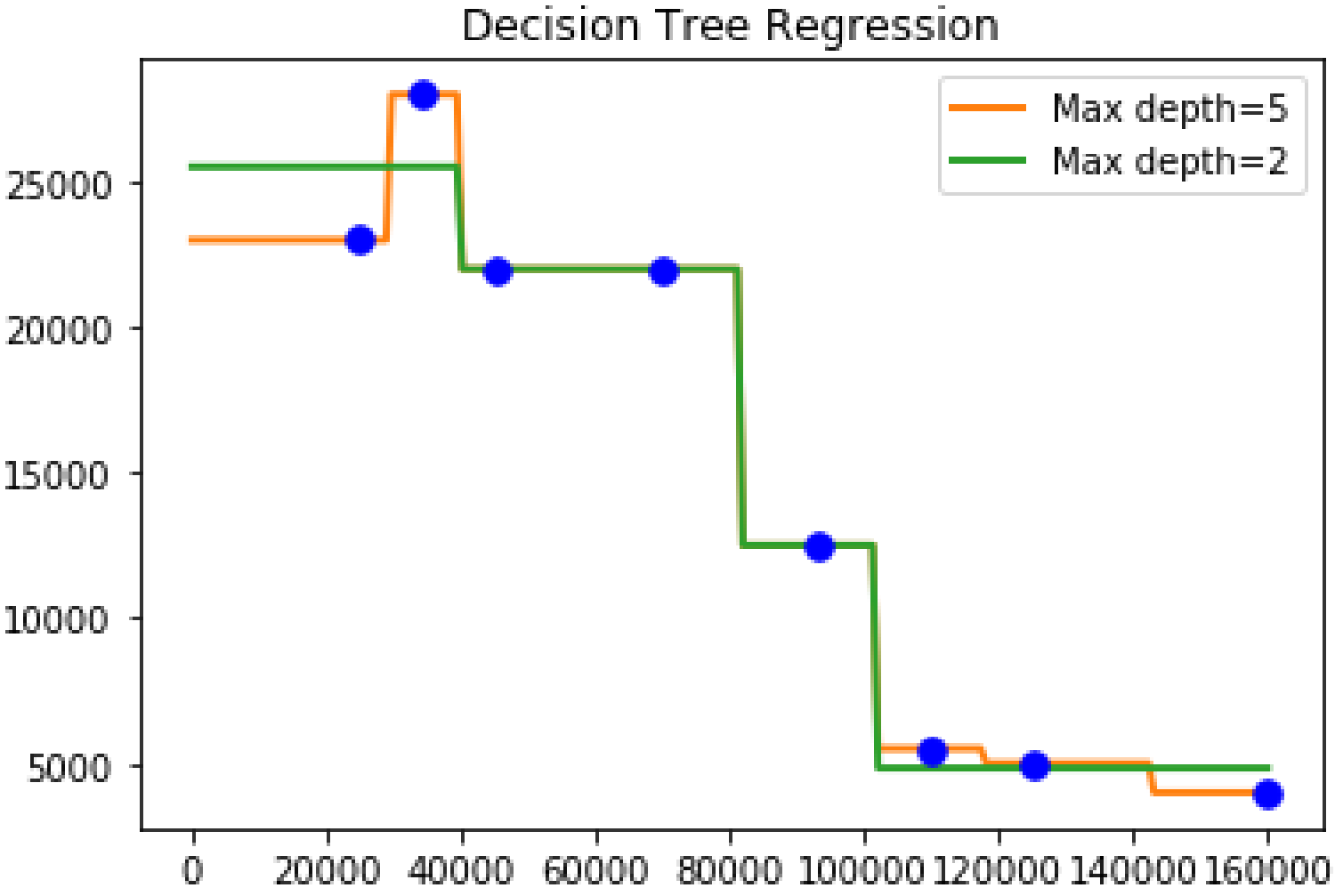
Nonparametric Regression

Decision Trees



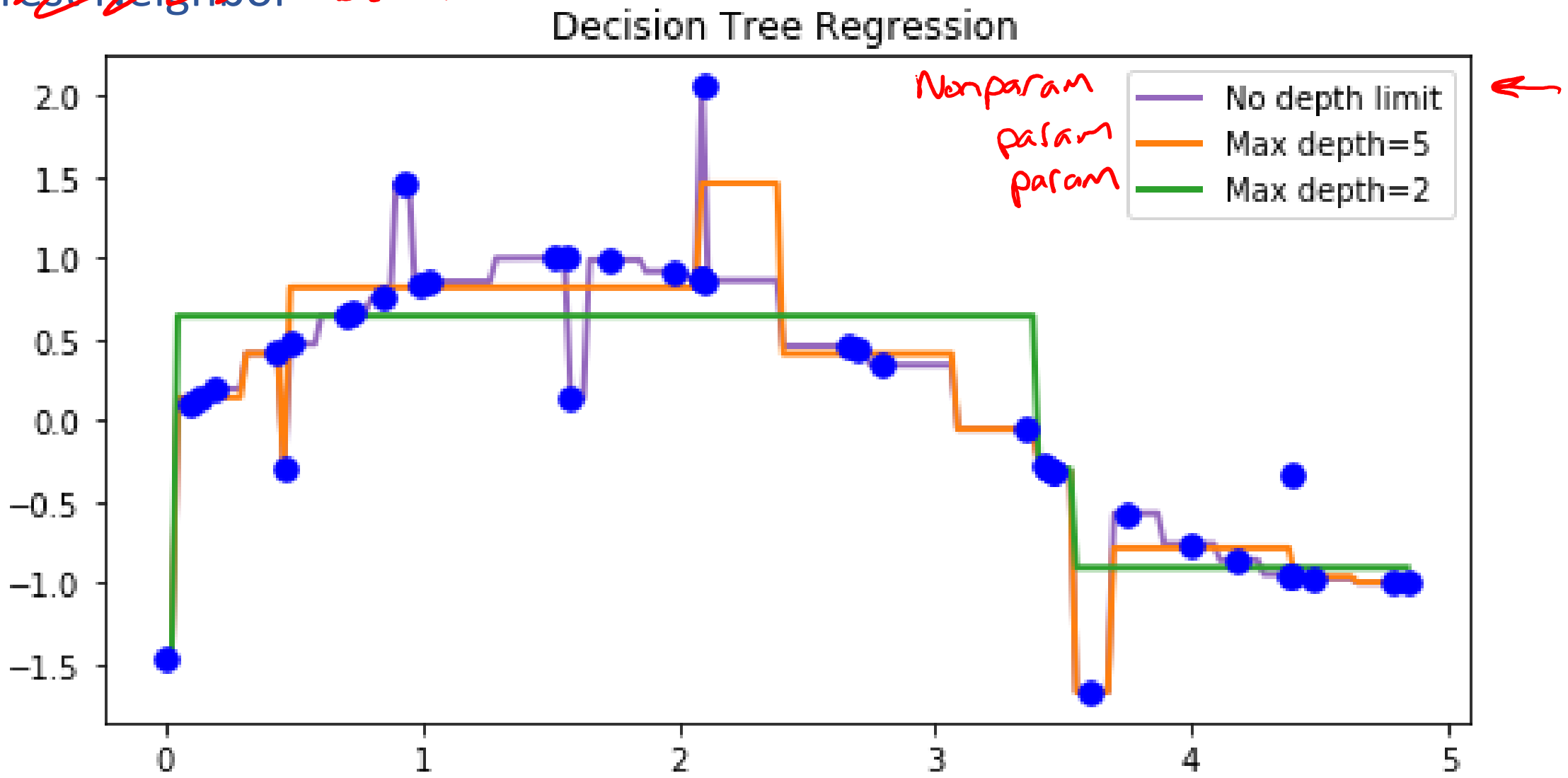
Nonparametric Regression

Decision Trees



Nonparametric Regression

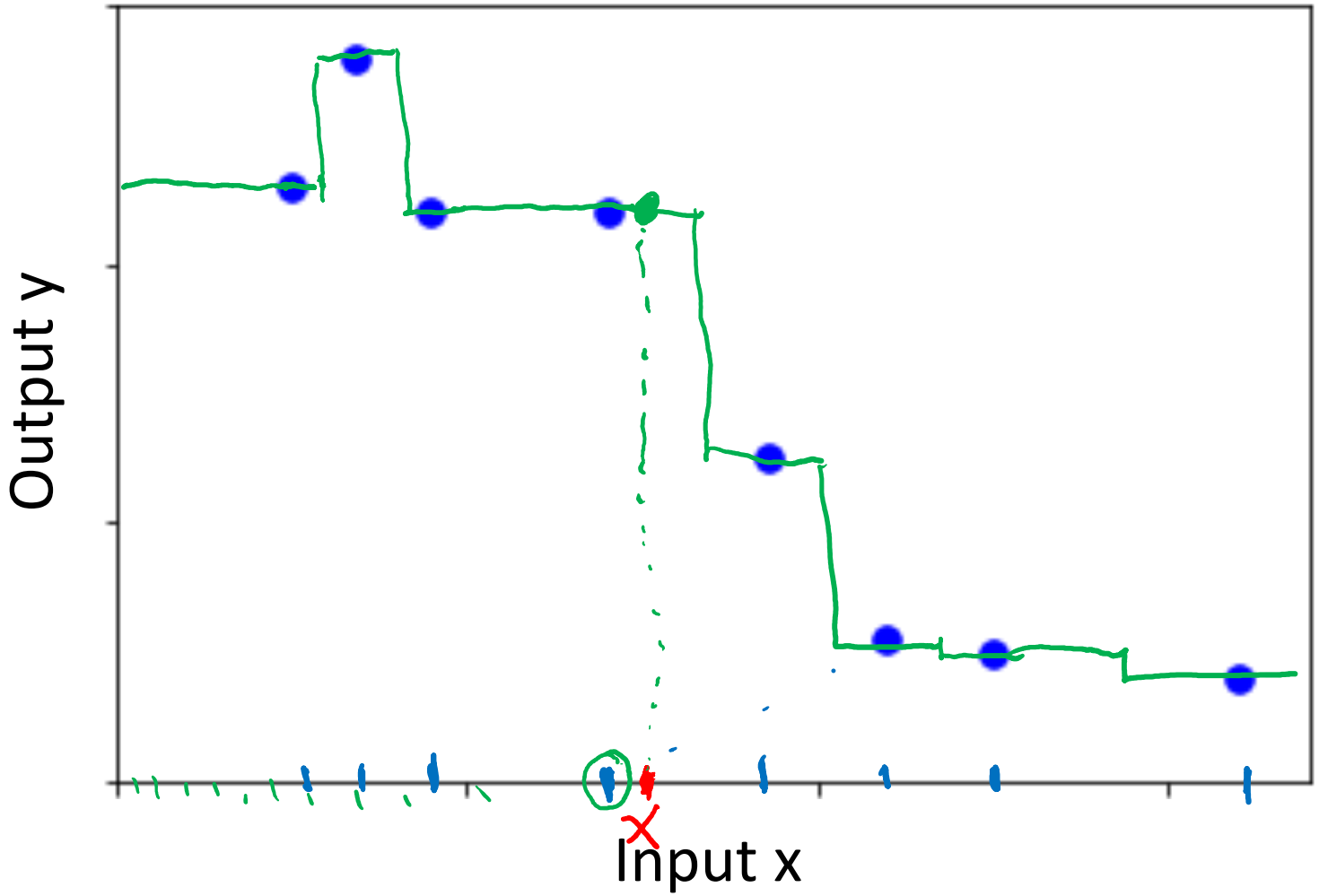
~~Nearest Neighbor~~ Decision Tree



Nonparametric Regression

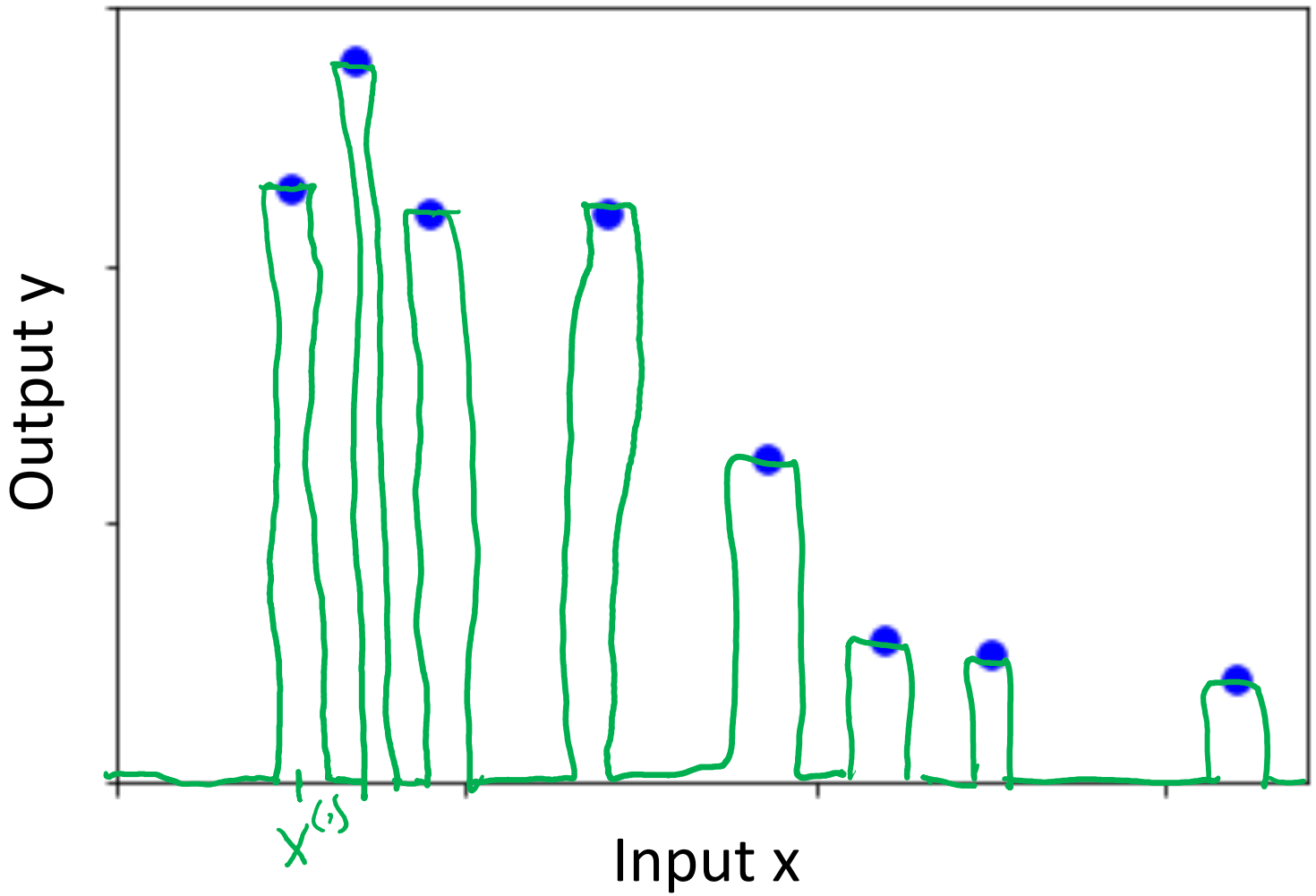
$\hat{y} = h(x)$

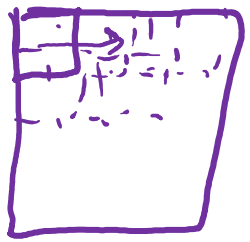
Nearest Neighbor

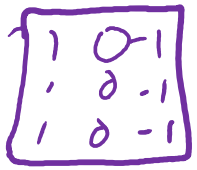


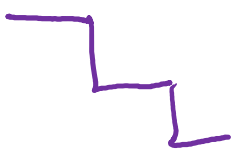
Nonparametric Regression

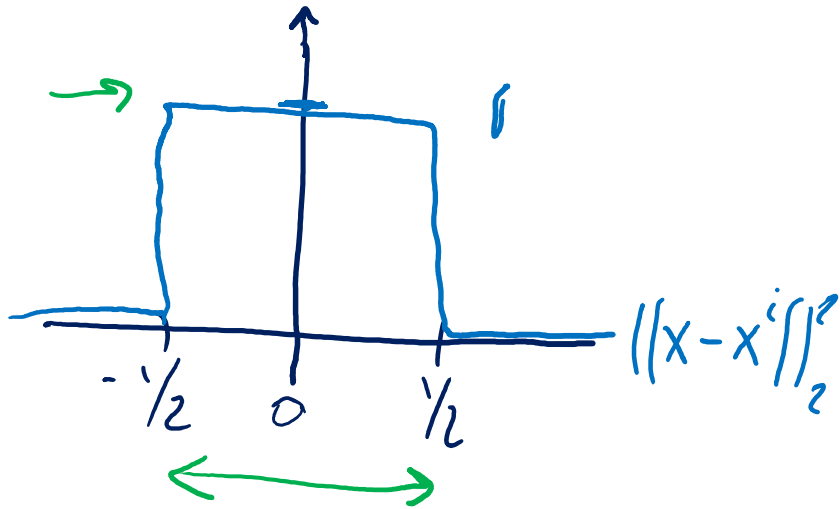
Kernel Regression









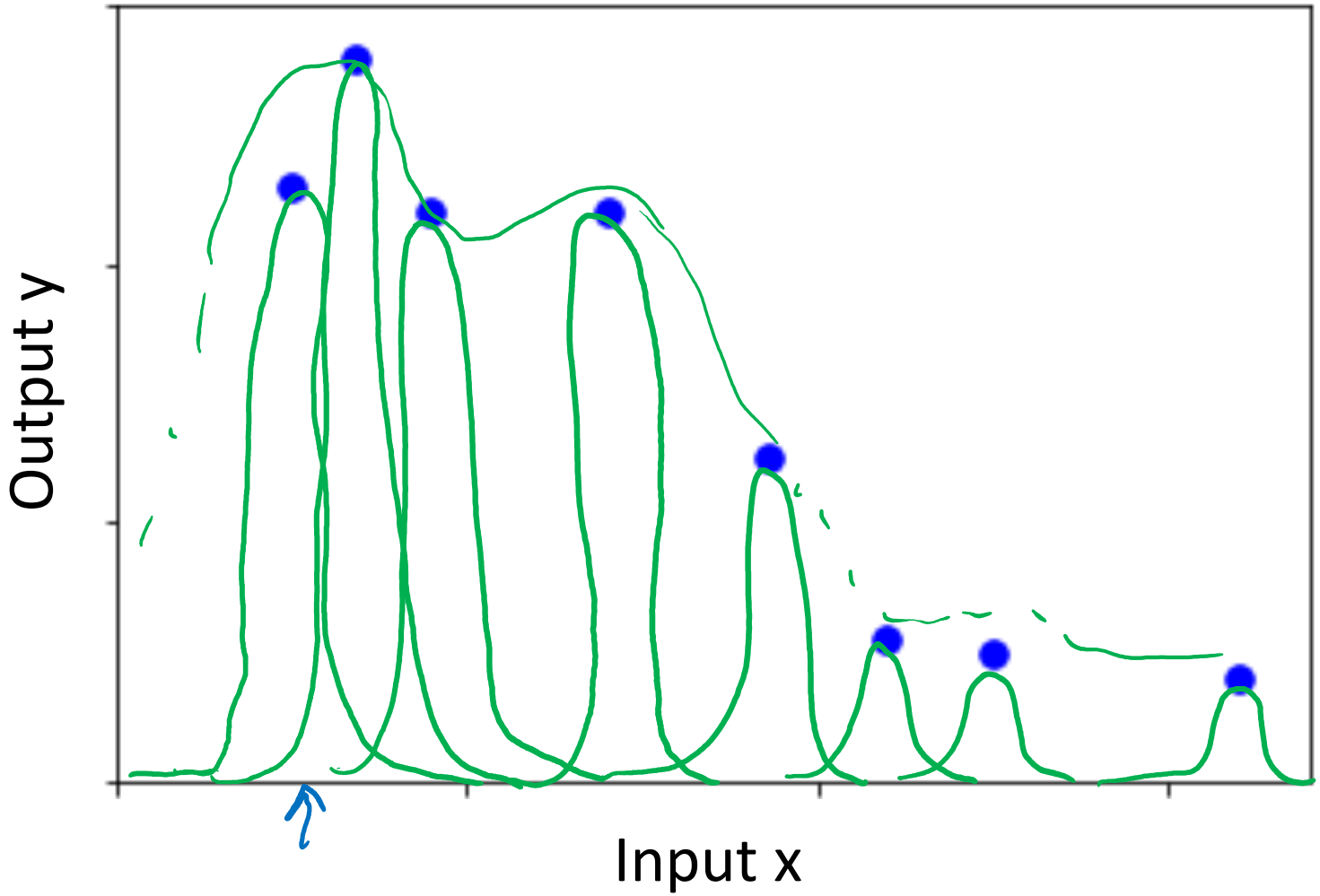


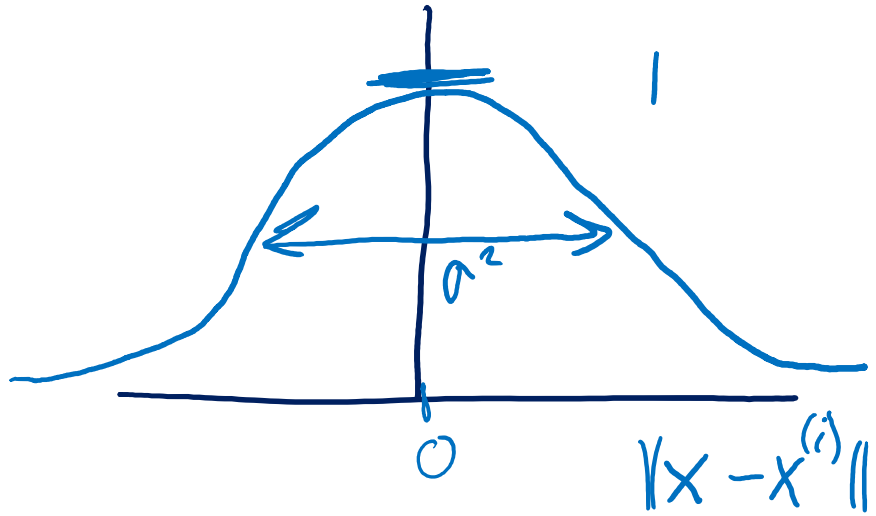
$$\hat{y} = h(x)$$

$$= \sum_i \alpha_i \, k(x, x^{(i)})$$

Nonparametric Regression

Kernel Regression



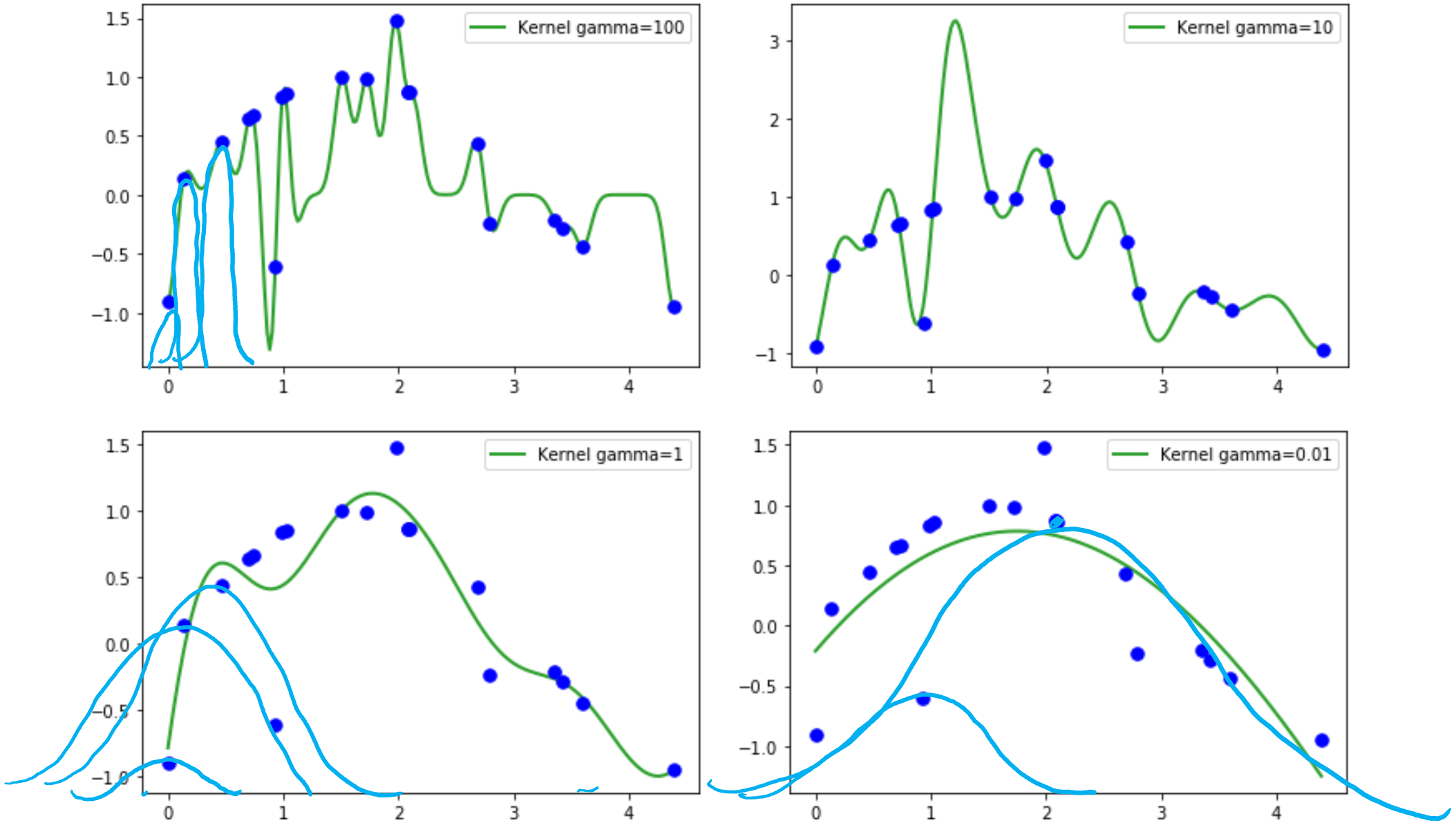


$$k(x, x') = e^{\frac{-\|x - x'\|_2^2}{2\sigma^2}}$$

$$= e^{-\gamma \|x - x'\|_2^2}$$

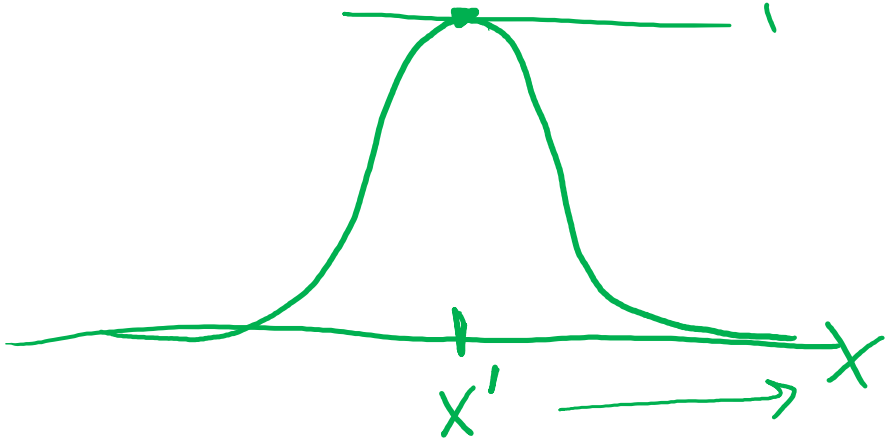
$$\gamma = \frac{1}{2\sigma^2}$$

Kernel Regression



Kernel Regression

RBF kernel and corresponding hypothesis function

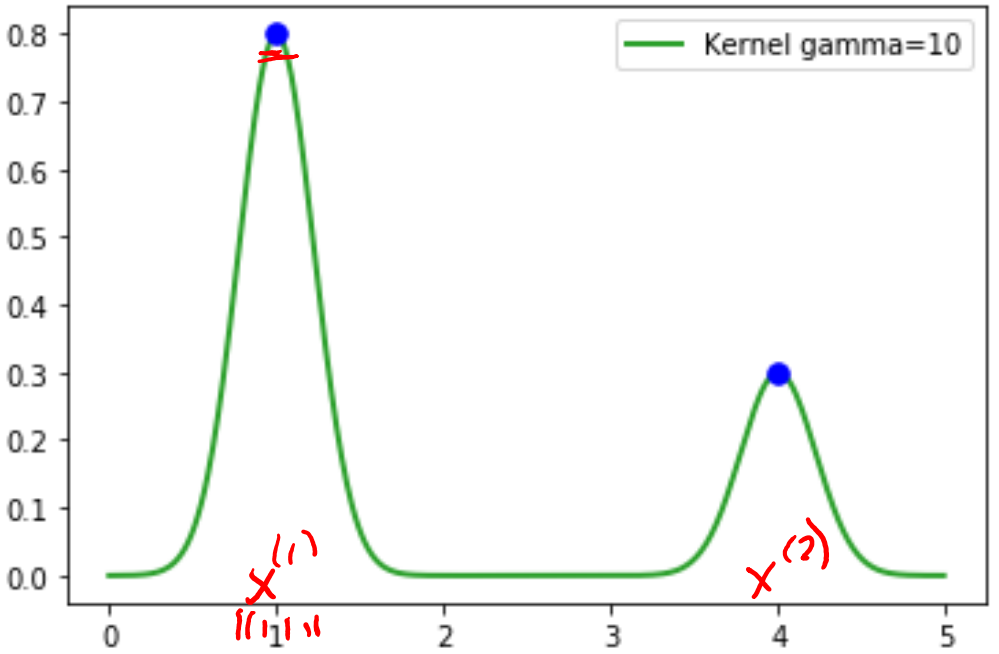


Distance kernel (Gaussian / Radial Basis Function)

- Close to point should be that point
- Far should be zero
- Mini Gaussian window

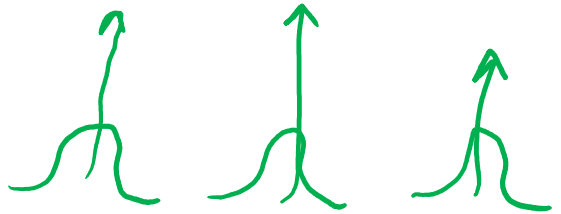
$$k(x, x') = e^{\frac{-\|x-x'\|_2^2}{2\sigma^2}} = e^{-\gamma\|x-x'\|_2^2}$$

- We control the variance



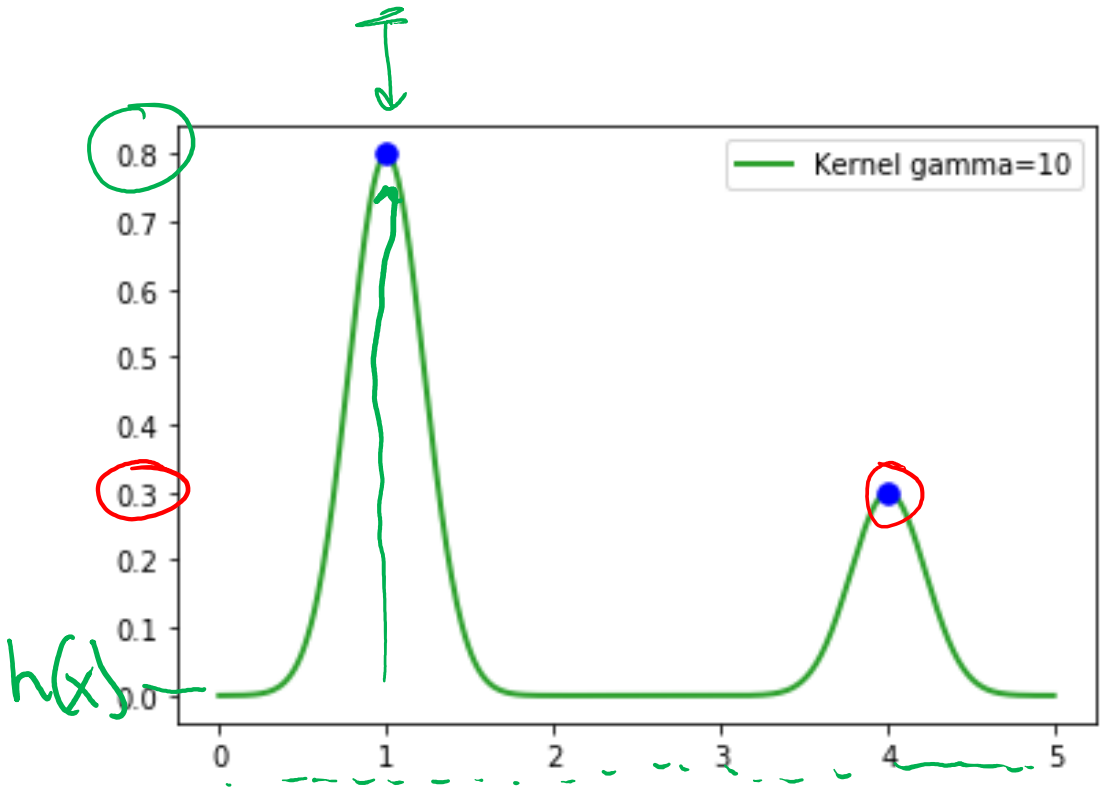
Kernel Regression

RBF kernel and corresponding hypothesis function



Prediction?

- Weighted sum of these little windows
  - $\hat{y} = h(x) = \sum_i \alpha_i k\left(x, x^{(i)}\right)$
  - What should α_i be?
    - $\alpha_i = y_i, \alpha = y$?
    - Need to account for points that are close together

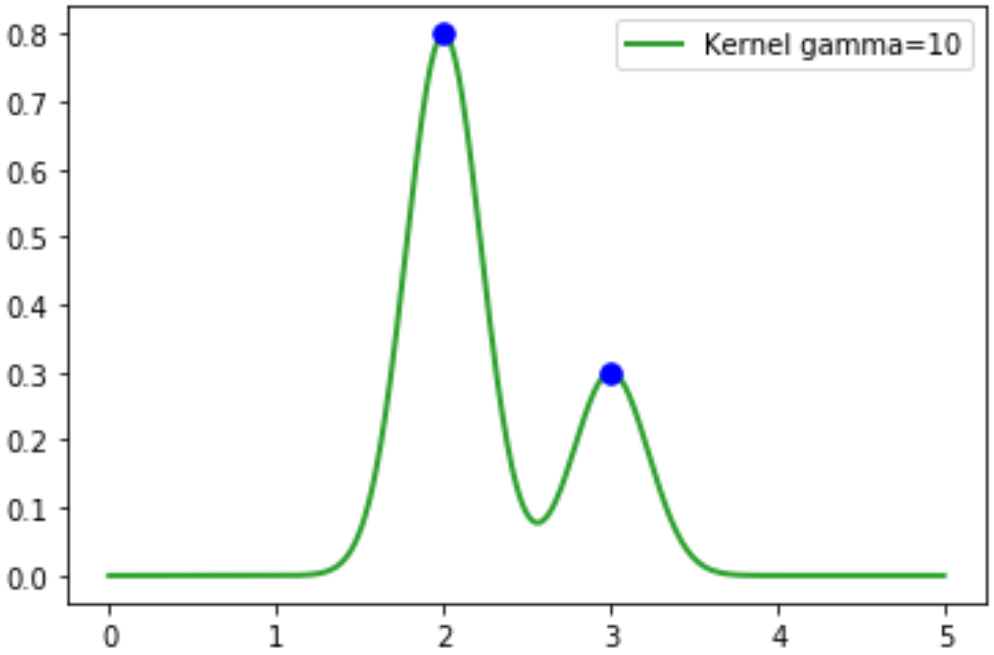


Kernel Regression

RBF kernel and corresponding hypothesis function

Prediction?

- Weighted sum of these little windows
  - $\hat{y} = h(x) = \sum_i \alpha_i k\left(x, x^{(i)}\right)$
  - What should α_i be?
    - $\alpha_i = y_i, \alpha = y$?
    - Need to account for points that are close together

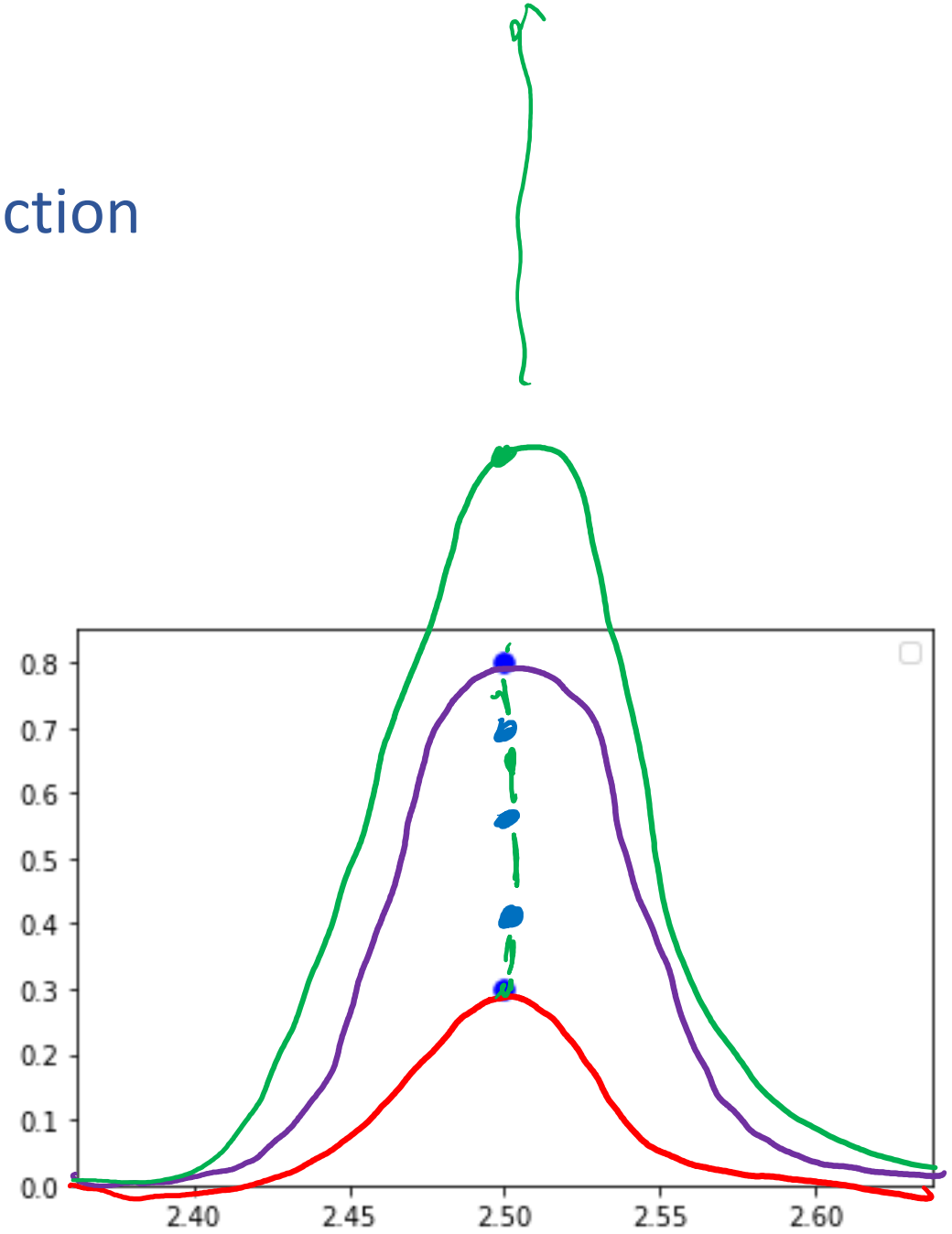


Kernel Regression

RBF kernel and corresponding hypothesis function

Prediction?

- Weighted sum of these little windows
  - $\hat{y} = h(x) = \sum_i \alpha_i k\left(x, x^{(i)}\right)$
  - What should α_i be?
  - $\alpha_i = y_i, \alpha = y$?
    - Need to account for points that are close together

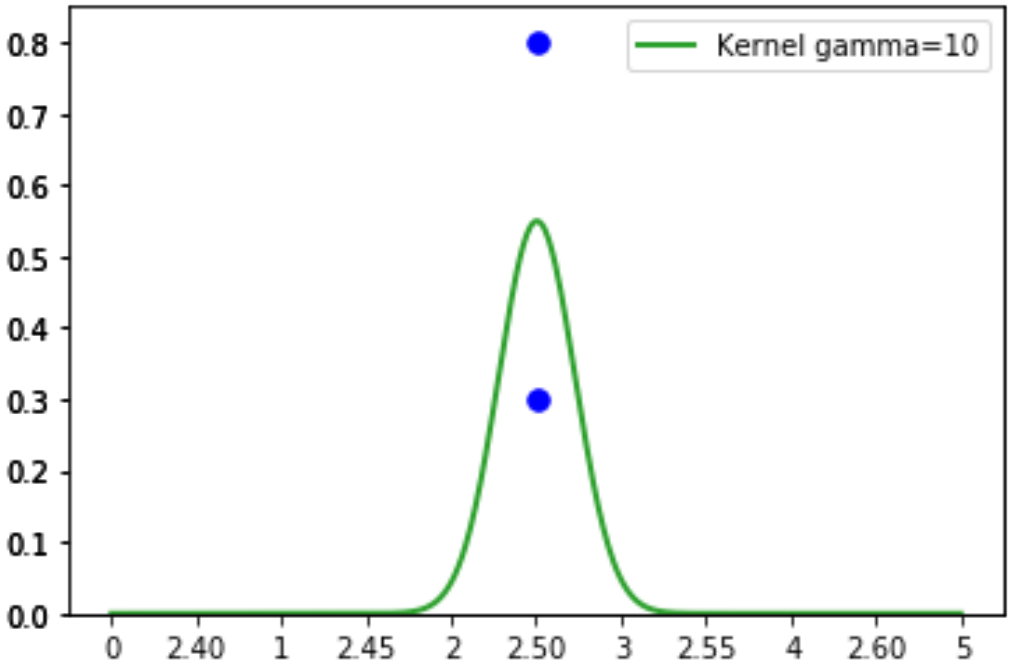


Kernel Regression

RBF kernel and corresponding hypothesis function

Prediction?

- Weighted sum of these little windows
  - $\hat{y} = h(x) = \sum_i \alpha_i k\left(x, x^{(i)}\right)$
  - What should α_i be?
    - Need to account for points that are close together



Kernel Regression

RBF kernel and corresponding hypothesis function

Prediction?

- Weighted sum of these little windows
  - $\hat{y} = h(x) = \sum_i \alpha_i k\left(x, x^{(i)}\right)$
  - What should α_i be?
    - Need to account for points that are close together

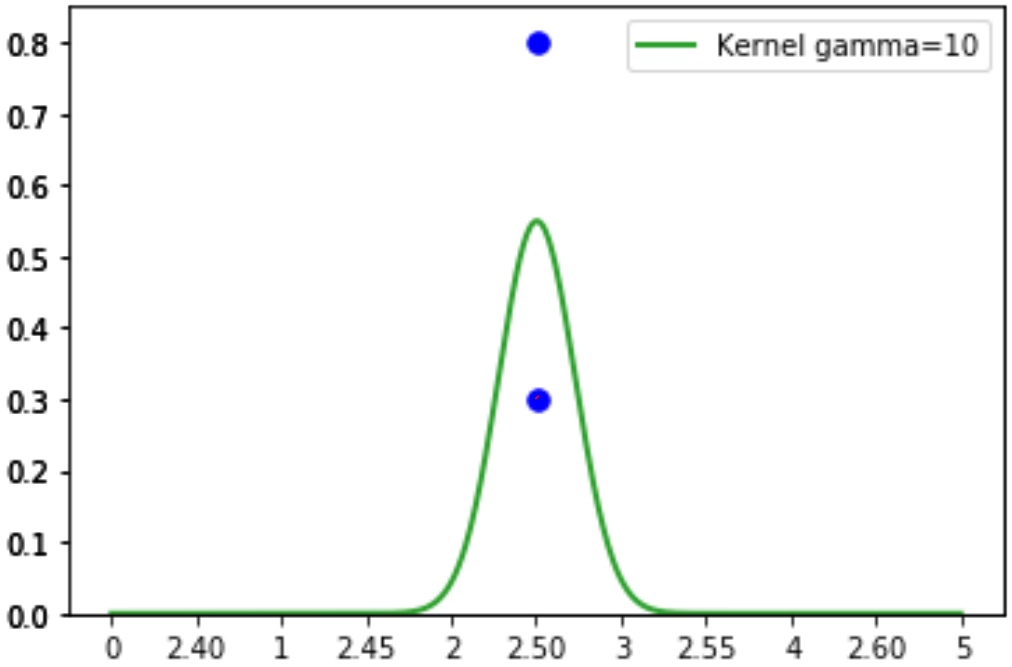
$$\hat{\alpha} = (K)^{-1}\hat{y}$$

$$\hat{\alpha} = (K + \lambda I)^{-1} y$$

where $K_{ij} = k\left(x^{(i)}, x^{(j)}\right)$

and λ is small to help inversion

$\alpha_i = \frac{y_i}{\text{density training}}$



Kernelized Linear Regression

Reminder: Polynomial Linear Regression

Polynomial feature function

Least squares formulation

Least squares solution

Reminder: Polynomial Linear Regression

Polynomial feature function

- $x \rightarrow \phi(x) = [1, x, x^2, x^3]^T$
- $X \rightarrow \Phi$

Least squares formulation

- $\min_w \|y - \Phi w\|_2^2$

Least squares solution

- $w = (\Phi^T \Phi)^{-1} \Phi^T y$

Plus L2 regularization

- $\min_w \|y - \Phi w\|_2^2 + \lambda \|w\|_2^2$
- $w = (\Phi^T \Phi + \lambda \mathrm{I})^{-1} \Phi^T y$

Can rewrite as

- $w = \Phi^T (\Phi \Phi^T + \lambda \mathrm{I})^{-1} y$

Kernelized Linear Regression

$\Phi \in R^{N \times (d+1)}$

$k(x, x') = \phi(x)^T \phi(x')$

L2 regularized linear regression (with feature function)

- $\min_w \|y - \Phi w\|_2^2 + \lambda \|w\|_2^2$
- $w = \left(\Phi^T \Phi + \lambda \mathrm{I}\right)^{-1} \Phi^T y$

Can rewrite as

- $w = \Phi^T \underbrace{\left(\Phi \Phi^T + \lambda \mathrm{I}\right)^{-1} y}_{\alpha} = \phi^T \alpha$

Let α ($N \times 1$)

$\boldsymbol{\alpha} = \left(\Phi \Phi^T + \lambda \mathrm{I}\right)^{-1} \mathbf{y}$

$\alpha = (K + \lambda I)^{-1} y$

$K_{ij} = k(x^{(i)}, x^{(j)})$

Prediction

- $\hat{y} = h(x) = \mathrm{w}^{\mathrm{T}} \mathrm{x}$ $\quad w^T \phi(x)$

$= \phi(x)^T w$

$= \phi(x)^T \phi^T \alpha$

$= \sum_i^N \alpha_i \phi(x)^T \phi(x^{(i)}) = \sum_i^N \alpha_i k(x, x^{(i)})$

$k(x, x') \in$ RBF