Artificial Intelligence Methods for Social Good

M3-3 [Machine Learning]:

Regression

08-537 (9-unit) and 08-737 (12-unit)
Instructor: Fei Fang
feifang@cmu.edu
Wean Hall 4126

Quiz 1: Recap: Gradient Descent

- Exp 1

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} \sum_{i=1}^{3} (y_i - (ax_i + b))^2$$
$$\text{s.t. } a, b \in \mathbb{R}$$

- $a^0 = 1, b^0 = 0.5$, run one iteration of gradient descent, we get
- A: $a^1 > a^0, b^1 < b^0$
- B: $a^1 > a^0, b^1 > b^0$
- C: $a^1 < a^0, b^1 < b^0$
- D: $a^1 < a^0, b^1 > b^0$

Recap

- Input x → (Not fully hard-coded) Program → Output y

- Key questions
  - Representation: How to represent the relationship between input and output
  - Inference: How to infer the output from input
  - Learning: How to learn the best model to describe the data?

Outline

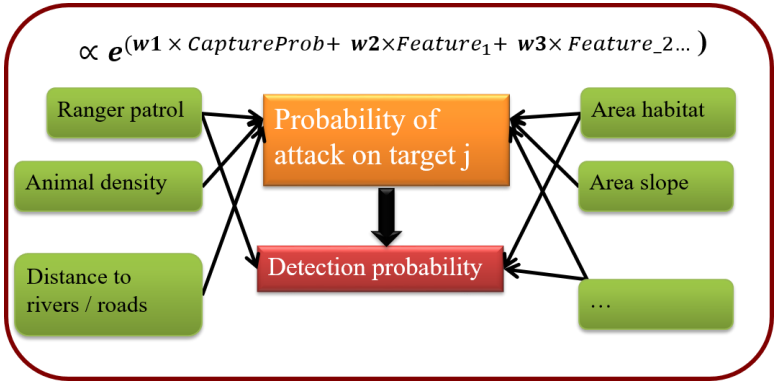
- What is Regression
- Parametric Regression
  - Simple Linear Regression
  - Linear Regression with Multiple Features
- Evaluation
  - Loss function
  - Train error, true error, test error
  - Overfitting
  - Model selection: Validation set
- Regularization
  - Ridge regression (L2)
    - Gradient descent
  - Lasso regression (L1)
    - Coordinate descent
  - Cross validation
- Non-parametric regression
  - K-Nearest Neighbors
  - Kernel Regression
- Logistic Regression

Learning Objectives

- Understand the concept of
  - Regression
  - Regularization
  - Cross validation
  - K-NN
  - Kernel regression
- Describe coordinate descent algorithm and can apply it to small scale problem
- Know how to find the algorithm/solver/package to do linear regression / kernel regression

What is Regression

- Continuous-valued output
- Exp 2: Predict house price
  - Input: size (sqft), #bedrooms, #bathrooms
  - Output: price of house
- Exp 3: Predict probability of detecting snares



- Exp 4: Predict #likes of a Tweet
  - Input: #followers, topic
  - Output: #likes
- Other examples?

Parametric Regression

- Data: input-output pairs (x_i, y_i)
- Representation: How to represent the relationship between input and output
  - Output is a function of input + noise, $y_i = f(x_i) + \epsilon_i$
  - Potential functions: constant, linear, quadratic, …
  - Which class of functions is more reasonable?
  - Within a class of functions (parameterized), which function is most reasonable?
  - Exp 1:

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$f(x) = c? \; f(x) = ax + b? \; f(x) = ax^2 + bx + c?$$

  - Exp 2: Predict house price

$$f(\text{housesize}, \#\text{bedrooms}, \#\text{bathrooms}) = a \cdot \text{housesize} + b \cdot \#\text{bedrooms} + \text{c} \cdot \#\text{bathrooms} + \text{d?}$$
$$f(\text{housesize}, \#\text{bedrooms}, \#\text{bathrooms}) = a \cdot \log(\text{housesize}) + b \cdot \#\text{bedrooms}^2 + \text{c} \cdot \#\text{bathrooms}^2 + \text{d?}$$

Parametric Regression

- Inference: How to infer the output from input
  - Estimate the output to be $f(x)$
  - Exp 1:

$$f(x) = 2x + 2$$

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

  - Exp 2: Predict the price of a house

$$f(\text{housesize}, \#\text{bedrooms}, \#\text{bathrooms}) = 100 \cdot \text{housesize} + 20 \cdot \#\text{bedrooms} + 10 \cdot \#\text{bathrooms} + 10000$$

Parametric Regression

- Learning: How to learn the best model to describe the data?
  - Given a class of functions (parameterized), find the best parameters
  - Solve an optimization problem that minimizes loss function
  - Example loss function - Residual sum of squares (RSS):
    - $RSS = \sum_i (y_i - f(x_i))^2$
  - Exp 1:

$$f(x) = ax + b$$

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} \sum_{i=1}^{3} (y_i - (ax_i + b))^2$$

$$\text{s.t. } a, b \in \mathbb{R}$$

Simple Linear Regression

- Assume
  - $f(x) = ax + b$
  - $y_i = ax_i + b + \epsilon_i$
- Easy to find the best parameter values when using RSS as the loss function
  - Approach 1: set gradient = 0, get closed-form solution
  - Approach 2: use gradient descent
- Exp 1:

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} \sum_{i=1}^{3} (y_i - (ax_i + b))^2$$

s.t. $a, b \in \mathbb{R}$

Quiz 2

- In Exp 1 of M3-3, what is the closed form representation for the optimal value of a if there are N data points? Explain how you get the formula.
  - A: $a = \sum_i x_i y_i \,/\, \sum_i x_i^2$
  - B: $a = \dfrac{\sum_i x_i y_i - \frac{\sum_i x_i \sum_i y_i}{N}}{\sum_i (x_i^2) - \frac{(\sum_i x_i)^2}{N}}$
  - C: $a = \dfrac{\sum_i x_i y_i - \frac{\sum_i x_i \sum_i y_i}{N}}{(\sum_i x_i)^2 - \frac{\sum_i x_i y_i}{N}}$
  - D: $a = \dfrac{\sum_i x_i^2 - \frac{\sum_i x_i y_i}{N}}{(\sum_i x_i)^2 - \frac{\sum_i x_i y_i}{N}}$

$$\min_{a,b} \sum_{i=1}^{N} (y_i - (ax_i + b))^2$$
$$\text{s.t. } a, b \in \mathbb{R}$$

Simple Linear Regression

- Exp 2: Ignore other input features
  - $f(\text{housesize}) = a \cdot \text{housesize} + b$
- Other cost functions?
  - E.g., higher cost for over-estimation in Exp 2 (no offer)

Linear Regression with Multiple Features

- What if

$$f(\text{housesize}) = w_0 + w_1 \cdot \text{housesize} + w_2 \cdot \text{housesize}^2?$$

$$f(\text{housesize}, \#\text{bedrooms}, \#\text{bathrooms}) = a \cdot \text{housesize} + b \cdot \#\text{bedrooms} + \text{c} \cdot \#\text{bathrooms} + \text{d}?$$

$$f(\text{housesize}, \#\text{bedrooms}, \#\text{bathrooms}) = a \cdot \log(\text{housesize}) + b \cdot \#\text{bedrooms}^2 + \text{c} \cdot \#\text{bathrooms}^2 \cdot \#\text{bedrooms}?$$

- Generalized model
  - $x = (x[1], x[2], \ldots, x[d])$: a vector of d dimensions
  - $h_j(x)$: a known function of x (no unknown parameters)
  - $f(x) = \sum_j w_j h_j(x)$
  - $y_i = f(x) + \epsilon_i$

Linear Regression with Multiple Features

- Residual sum of squares (RSS):
  - $RSS = \sum_i (y_i - f(x_i))^2$

- Find the best parameter values
  - Write down the optimization problem
  - Solve the optimization problem
    - Approach 1: Set gradient = 0 → closed form solution
      - Rewrite in matrix form
      - Computational challenge: compute the inverse of a matrix can be time consuming! ($O(D^3)$)
    - Approach 2: Gradient descent
      - More general, Can be more efficient

Evaluation

- Access performance
  - Make predictions/estimations with the trained model at input x
  - Ideally: $y = f(x)$
  - Measure loss in non-ideal case: define a loss function
    - Absolute loss: $L = |y - f(x)|$
    - Squared loss: $L = \big(y - f(x)\big)^2$

Evaluation

- On which data points do you access the performance?
  - Training error
    - Average loss on all points in training set
    - Overly optimistic
  - True error
    - Average loss on all possible points weighted by likelihood
    - You don't know all points and how likely they are
  - Test error
    - Average loss on a set of points that are not used to train the model
    - Approximate true error
- Overfitting: low training error, high true error

Regression with Regularization

- Balance fitness to training data and model complexity to avoid overfitting

- Total cost = measure of fit + measure of model complexity

Ridge Regression (L2 Regularization)

- When model is complex, some trained parameter values would be very high

- Ridge total cost = measure of fit + measure of magnitude of coefficients = $RSS(w) + \lambda \|w\|_2^2$
  - λ: control the balance

Ridge Regression (L2 Regularization)

- Exp 1:

$$f(x) = ax + b$$

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} \sum_{i=1}^{3} (y_i - (ax_i + b))^2 + \lambda(a^2 + b^2)$$

$$\text{s.t. } a, b \in \mathbb{R}$$

Ridge Regression (L2 Regularization)

- How to choose λ?
  - Validation set

Training set	Validation set	Test set
80%	10%	10%

  - K-fold cross validation (Typically, $K = 5$ or 10)
    - Randomly assign data to K groups
    - Train on data from K-1 groups and compute error on remaining group (validation set)
    - Compute average error $CV(\lambda)$ for difference choice of validation set
    - Choose λ that minimizes $CV(\lambda)$

Lasso Regression (L1 Regularization)

- Lasso total cost = measure of fit + measure of magnitude of coefficients = $RSS(w) + \lambda \|w\|_1$
  - λ: control the balance

- Lead to sparse solutions in practice (less features with $w > 0$)
- Indicate which features are more important
- Trading off efficiency with interpretability

Lasso Regression (L1 Regularization)

- Exp 1:

$$f(x) = ax + b$$

x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} \sum_{i=1}^{3} (y_i - (ax_i + b))^2 + \lambda(|a| + |b|)$$

$$\text{s.t. } a, b \in \mathbb{R}$$

- If $a = 0, b = 3$, how to apply one step of gradient descent?
  - How to compute the gradient?

- Use coordinate decent instead of gradient descent!

Coordinate descent algorithm

- In every iteration, pick a coordinate j, fix the values in all other coordinates and pick the best value in j^{th} coordinate
  - To pick the best value in j^{th} coordinate: solve an optimization problem
    - Compute gradient
    - Set gradient = 0

Coordinate descent algorithm for Lasso Regression

- Initialize w
- While not converged
  - For j=0..D
    - Compute $z_j = \sum_i h_j(x_i)^2$
    - Compute $\rho_j = \sum_i h_j(x_i)(y_i - \hat{y}_i(w_{-j}))$ where $\hat{y}_i(w_{-j}) = \sum_{k \neq j} w_k h_k(x_i)$
    - Update w_j as
      - $(\rho_j + \frac{\lambda}{2})/z_j$ if $\rho_j < -\frac{\lambda}{2}$
      - 0 if $-\frac{\lambda}{2} \leq \rho_j < \frac{\lambda}{2}$
      - $(\rho_j - \frac{\lambda}{2})/z_j$ if $\rho_j > \frac{\lambda}{2}$

Coordinate descent algorithm for Lasso Regression

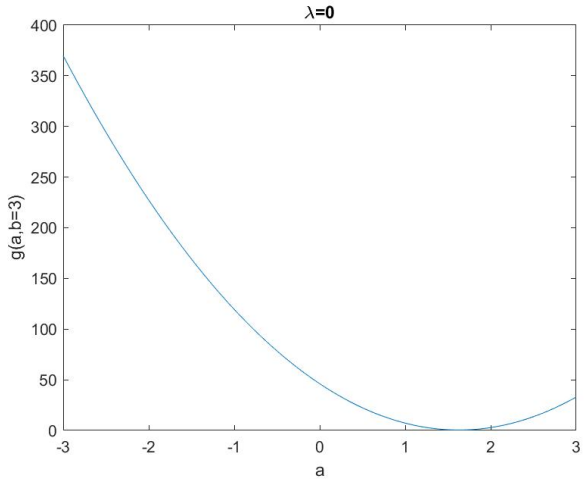
- Exp 1:
- Initialize $a = 0, b = 3$
- Apply coordinate descent
- Step 1: Update a
  - Check $g(a, b = 3)$

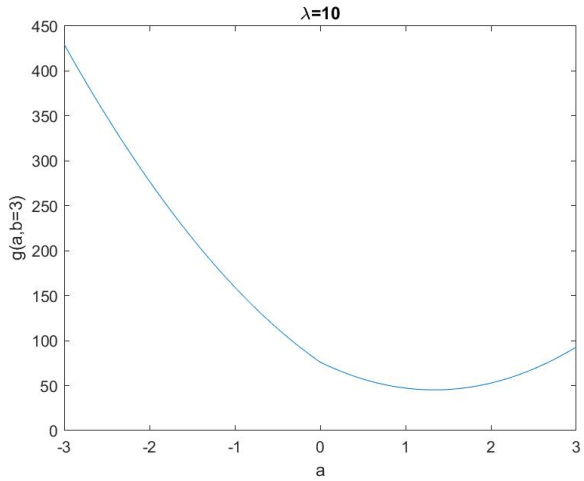
$$f(x) = ax + b$$

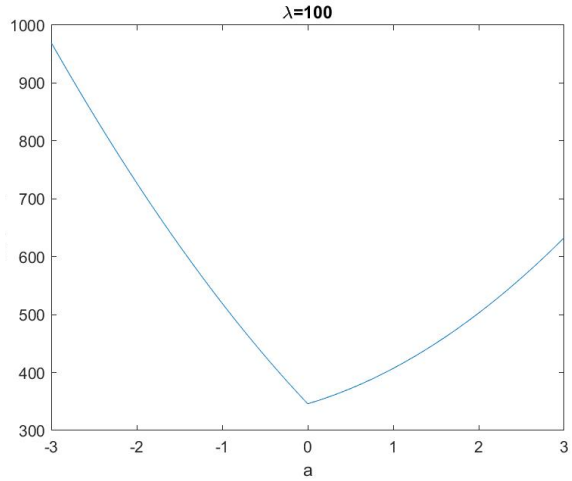
x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} g(a,b) = \sum_{i=1}^{3} (y_i - (ax_i + b))^2 + \lambda(|a| + |b|)$$

s.t. $a, b \in \mathbb{R}$







Coordinate descent algorithm for Lasso Regression

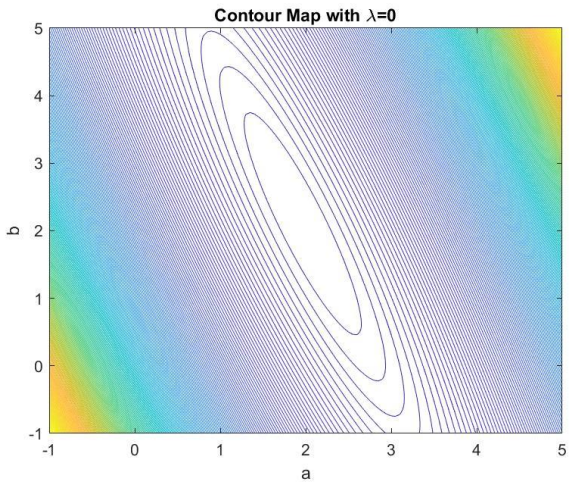
- Exp 1:
- Initialize $a = 0, b = 3$
- Apply coordinate descent
- Step 1-n: Update a, $b, a, b, \ldots$
  - Check contour map of $g(a, b)$

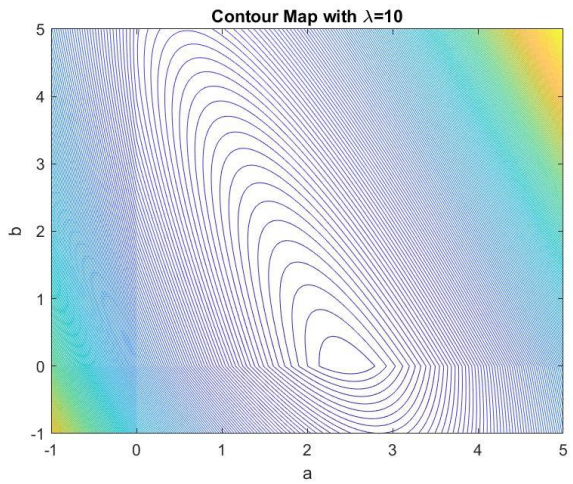
$$f(x) = ax + b$$

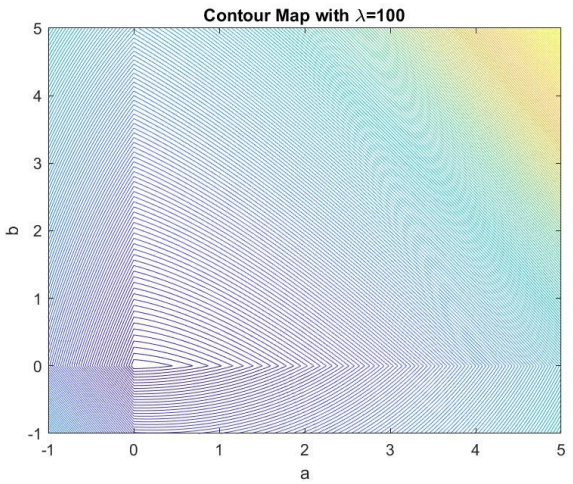
x_i	1.0	2.0	3.5
y_i	4.1	5.98	9.0

$$\min_{a,b} g(a,b) = \sum_{i=1}^{3} (y_i - (ax_i + b))^2 + \lambda(|a| + |b|)$$

s.t. $a, b \in \mathbb{R}$







How to handle intercept parameter

- Motivation of using Ridge or Lasso regression: get smaller parameter value so as to avoid overfitting
- Recall $f(x) = \sum_j w_j h_j(x)$. Often $h_0(x) = 1, w_0$ is the intercept parameter
- Ridge regression cost = $RSS(w) + \lambda \|w\|_2^2$: requires w_0 to be small. However, a large value of w_0 does not indicate overfitting!
- If y value of a dataset is shifted by M, the learned model should simply shift by M. But Ridge regression penalizes this shift heavily.

How to handle intercept parameter

- Solution 1: (used by many packages, e.g., sklearn)
  - Exclude intercept term in the loss function
  - Use modified Ridge regression cost = $RSS(w_0, w_{j\neq 0}) + \lambda \|w_{j\neq 0}\|_2^2$
- Solution 2: Shift the data first so that $\bar{y} = 0$, therefore $w_0 = 0$ for the shifted dataset. Train a model using standard ridge regression, and then shift back.

Outline

- What is Regression
- Parametric Regression
  - Simple Linear Regression
  - Linear Regression with Multiple Features
- Evaluation
  - Loss function
  - Train error, true error, test error
  - Overfitting
  - Model selection: Validation set
- Regularization
  - Ridge regression (L2)
    - Gradient descent
  - Lasso regression (L1)
    - Coordinate descent
  - Cross validation
- Non-parametric regression
  - K-Nearest Neighbors
  - Kernel Regression
- Logistic Regression

Non-parametric regression

- If we don't want to represent the relationship between x and y using an explicit function with to-be-learned parameters, can we still make assumptions?

- Simplest approach: Nearest neighbor regression

Nearest Neighbor Regression

- 1-NN regression
  - Predicted value = value of "closest" point in training data
  - Distance metric: (Scaled) Euclidean distance, Manhattan
  - Limitation: poor performance in areas with little data; sensitive to noise

- K-NN regression
  - Predicted value = average value of k "closest" point in training data
  - Robust to noise
  - Limitation: poor performance in areas with little data or boundary; discontinuous predictions
  - Choose k: cross validation

Quiz 3

- Given the following training data, what is the price of Alice's house with house size = 1300 sqft through k-NN with k=3
  - 220
  - 235
  - 213.3
  - 256.7

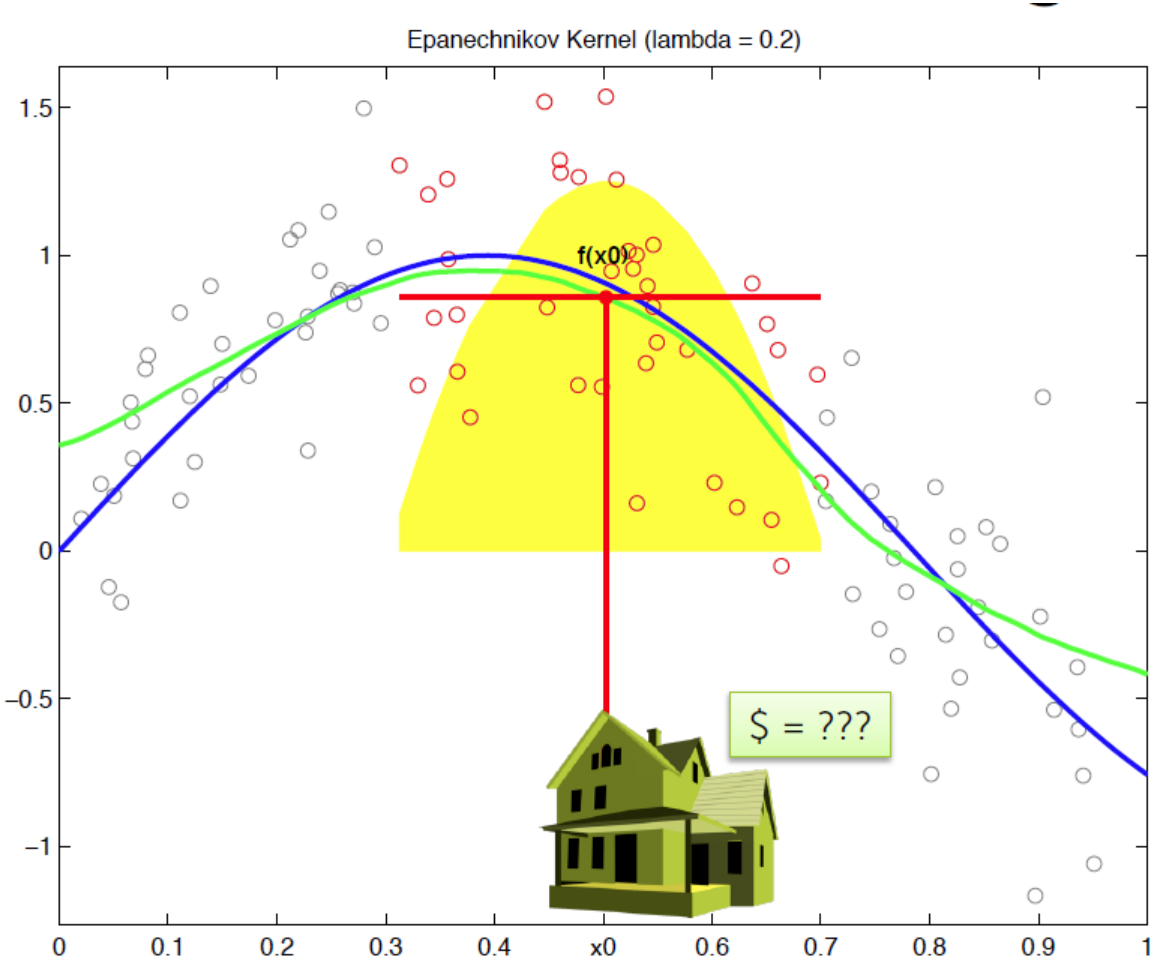
House size (sqft)	Sale price ($)
1200	220
1000	170
800	150
1500	250
1800	300

Kernel Regression

- Weighted K-NN regression
  - Predicted value = weighted average value of k "closest" point in training data
  - Smaller distance → Higher weight

- Kernel regression
  - Predicted value = weighted average value of all points in training data
  - Weight is a function of distance: kernel
  - Example kernel: Gaussian, Triangle, Uniform
    - Gaussian: $kernel_\lambda(|x_i - x_q|) = e^{-\frac{|x_i - x_q|^2}{\lambda}}$
  - Choose bandwidth parameter λ: cross validation

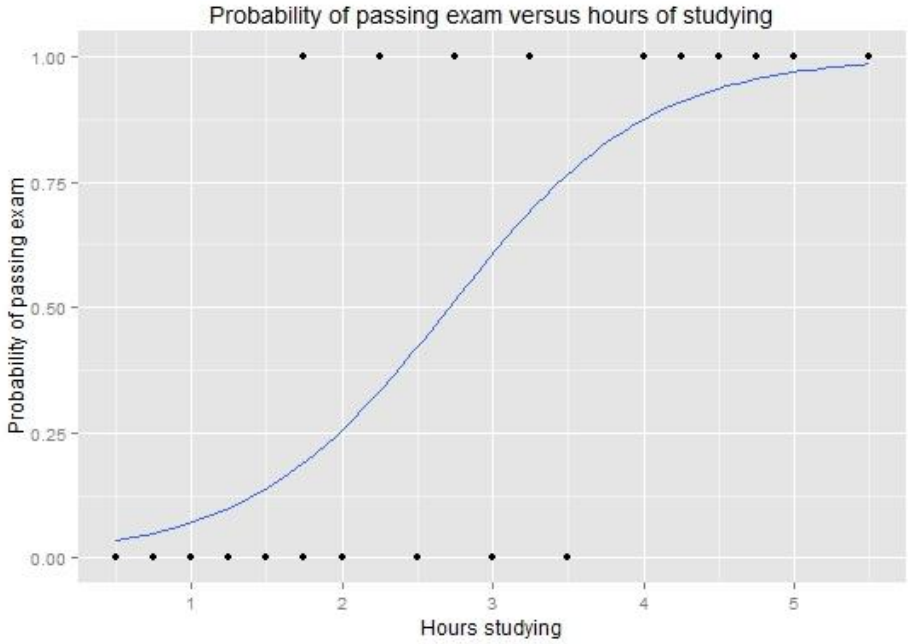
Kernel Regression



https://www.coursera.org/learn/ml-regression

Logistic Regression

- When the output is binary (or categorical), but we want to predict the probability of having positive label
- $p(y=1) = \frac{1}{1+e^{\sum_j w_j h_j(x)}}$



Tips for Regression in Practice

- Normalize the feature values
- Center the data
- Packages: sklearn (Python), fitlm(MATLAB)

Additional Resources

- Text book
  - *Pattern Recognition and Machine Learning*, *Chapters 3,6*
  - Christopher Bishop
- Online course
  - https://www.coursera.org/learn/ml-regression
  - https://www.coursera.org/learn/ml-classification
  - https://www.coursera.org/learn/machine-learning