DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 1 2020

Announcements

- Thanks for submitting HW 1
- HW 2 is posted on Piazza and Gradescope
  - It is due on Monday, Oct. 9, at midnight
- Start thinking about class projects
  - Will post guidelines soon
  - Find a project partner and dataset you are interested in

Outline

- Gradient Descent comparison with closed-form solution
- Regularization
  - Ridge and Lasso regression
- Classification
  - K Nearest Neighbors (kNN)
  - Cross-validation

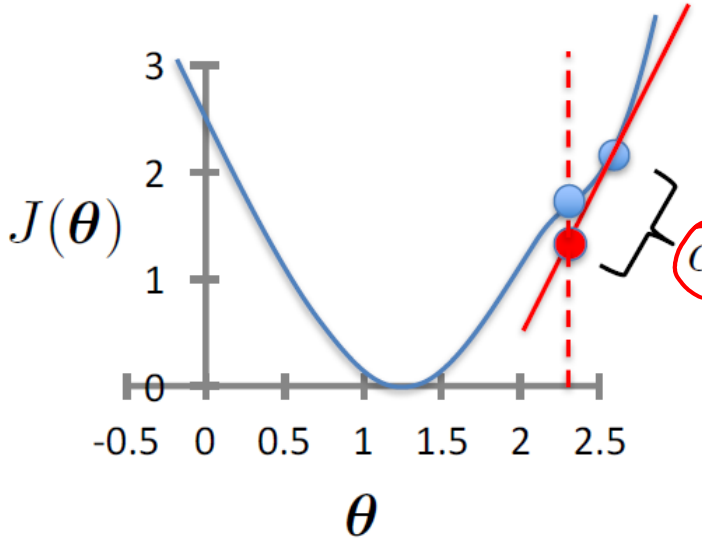
Gradient Descent

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

simultaneous update
for j = 0 ... d

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$



STEP SIZE

LEARNING RATE

STOPPING:

- NUMBER STEPS
- HOW MUCH θ CHANGE

GD for Linear Regression

- Initialize θ
- Repeat until convergence $||\theta_{new} - \theta_{old}|| < \epsilon$ or iterations == MAX_ITER

$$\theta_j \leftarrow \theta_j - \alpha \frac{2}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

simultaneous update for j = 0 ... d

- To achieve simultaneous update
  - At the start of each GD iteration, compute $h_\theta(x_i)$
  - Use this stored value in the update step loop
- Assume convergence when $\|\boldsymbol{\theta}_{new} - \boldsymbol{\theta}_{old}\|_2 < \epsilon$

L_2 norm: $$\|\boldsymbol{v}\|_2 = \sqrt{\sum_i v_i^2} = \sqrt{v_1^2 + v_2^2 + \ldots + v_{|v|}^2}$$

Can also bound number of iterations

Gradient Descent in Practice

- Asymptotic complexity
  - N is size of training data, d is feature dimension, and T is number of iterations

$O(NdT)$

- Most popular optimization algorithm in use today
- At the basis of training
  - Linear Regression
  - Logistic regression
  - SVM
  - Neural networks and Deep learning
  - Stochastic Gradient Descent variants

Gradient Descent vs Closed Form

Gradient Descent

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

Closed form

$$\boldsymbol{\theta} = (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y}$$

- Gradient Descent
  - + PERFORMANCE
  - + MEMORY
  - + GENERALIZABLE
  - + GLOBAL OPT. FOR CONVEX OBJ.
  - − NO GUARANTEES ON OPT.
  - − PARAM TUNING
- Closed Form
  - + GLOBAL OPTIMUM
  - + NO PARAM TUNING
  - − NOT ALWAYS INVERTIBLE

Issues with Gradient Descent

- Might get stuck in local optimum and not converge to global optimum
  - Restart from multiple initial points
- Only works with differentiable loss functions
- Small or large gradients
  - Feature scaling helps
- Tune learning rate
  - Can use line search for determining optimal learning rate

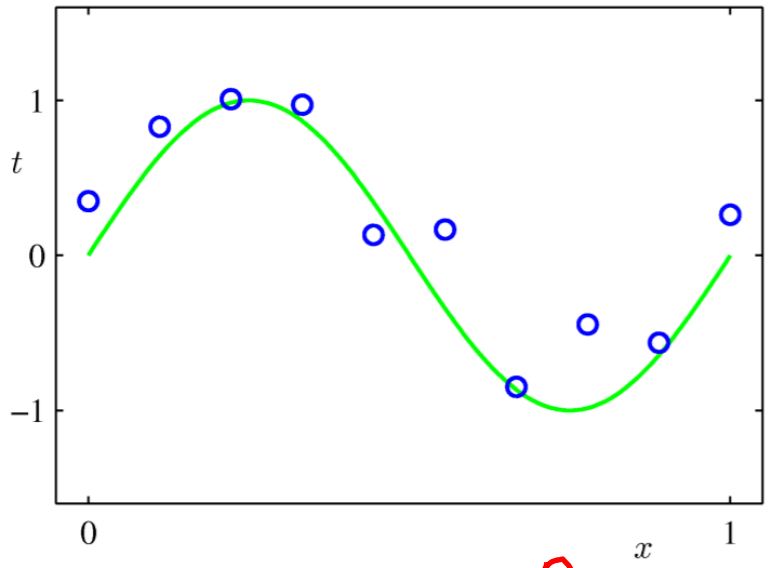
Review Gradient Descent

- Gradient descent is an efficient algorithm for optimization and training ML models
  - The most widely used algorithm in ML!
  - Much faster than using closed-form solution for linear regression
  - Main issues with Gradient Descent is convergence and getting stuck in local optima (for neural networks)
- Gradient descent is guaranteed to converge to optimum for strictly convex functions if run long enough

Polynomial Regression

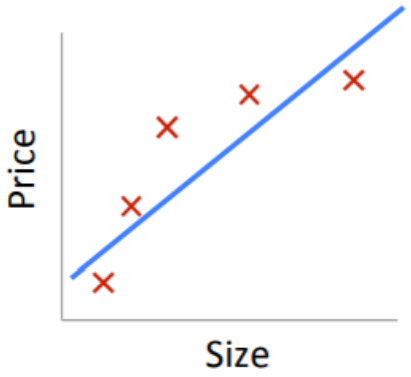
- Polynomial function on single feature
  - $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_d x^p$

degree P



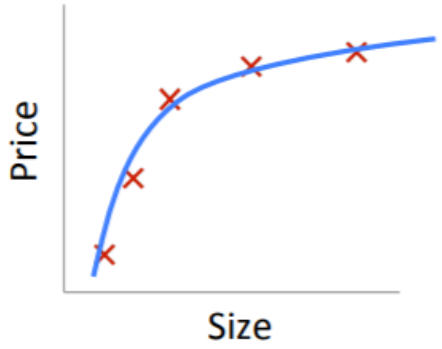
$x \rightarrow x, x^2, \ldots, x^P$

Polynomial Regression



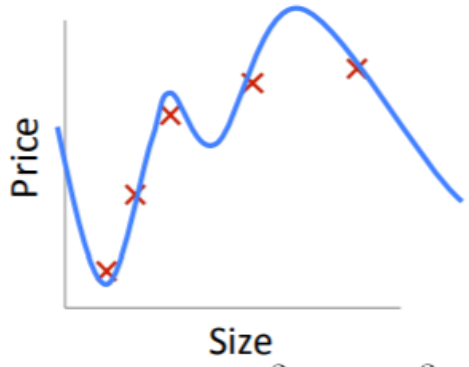
$\theta_0 + \theta_1 x$

UNDERFIT



$\theta_0 + \theta_1 x + \theta_2 x^2$

CORRECT FIT



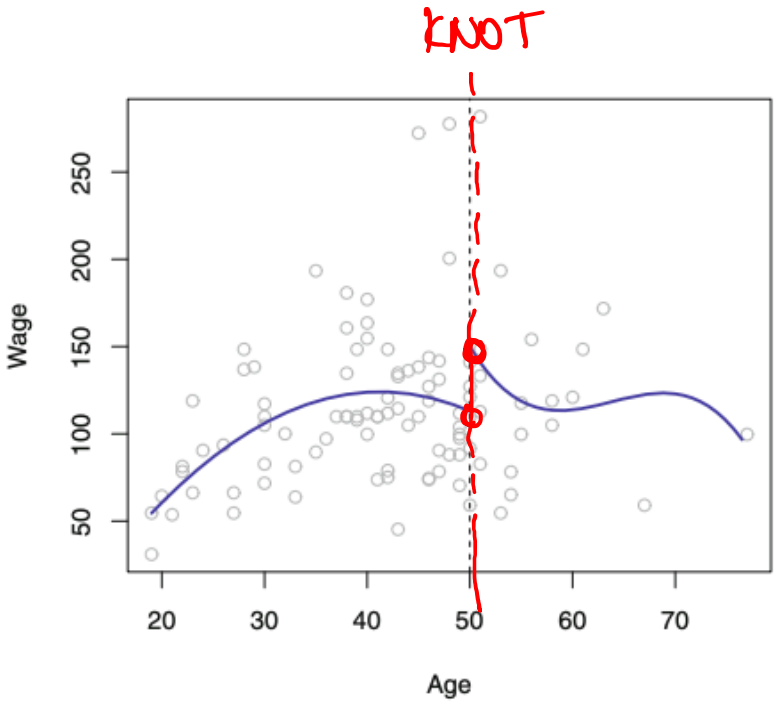
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

OVERFIT

TOO COMPLEX

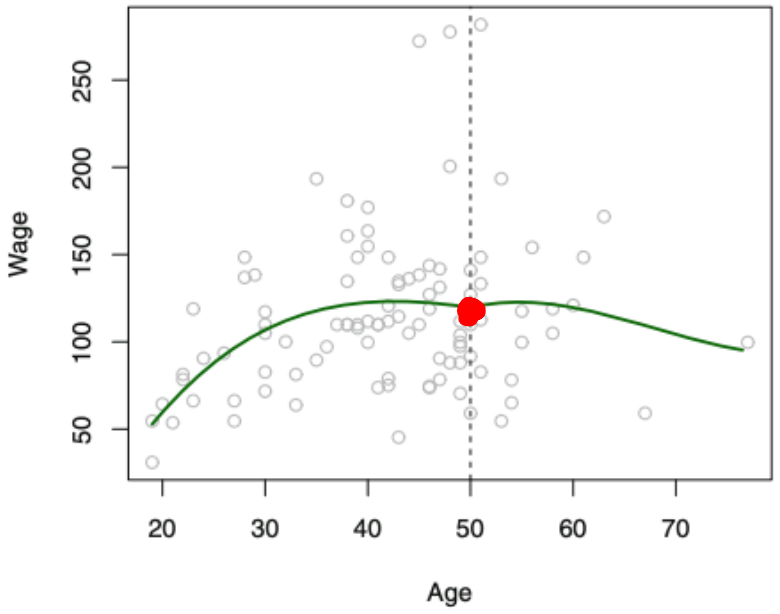
TO AVOID OVERFITTING: $R \leq 4$.

Non-Linear Regression



PIECEWISE CUBIC

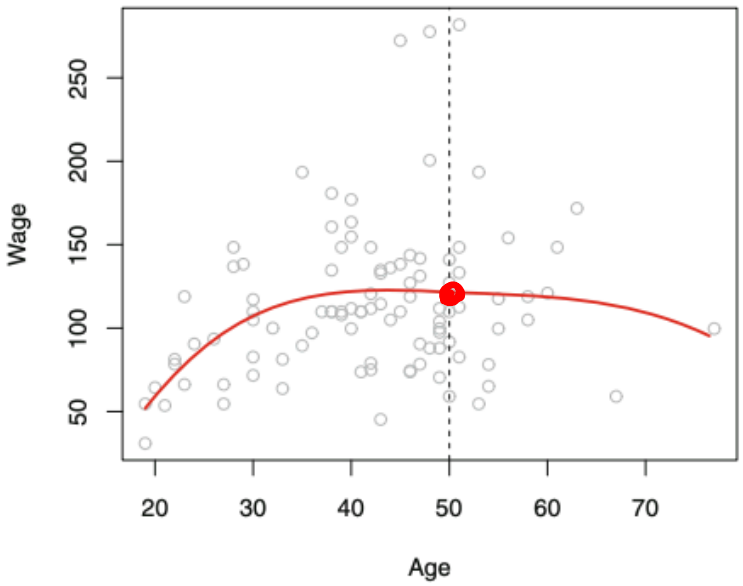
DEGS



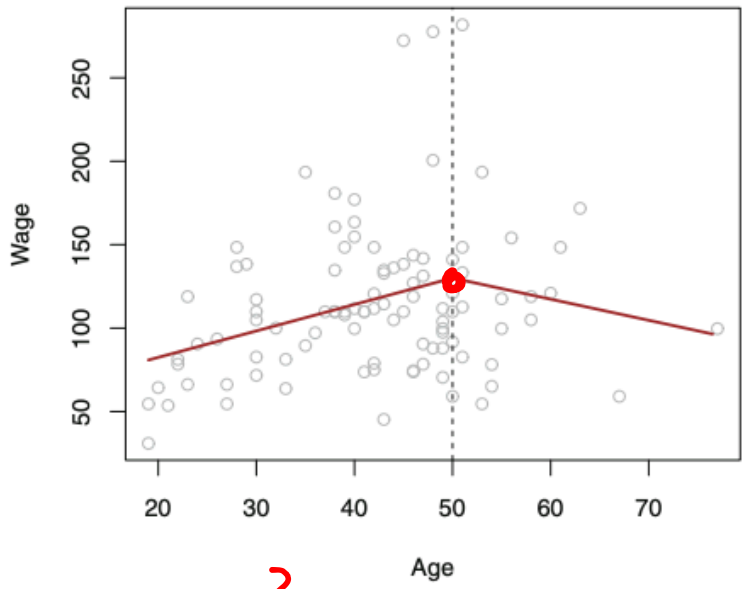
CONTINOUS ...

Spline Regression

CUBIC SPLINE



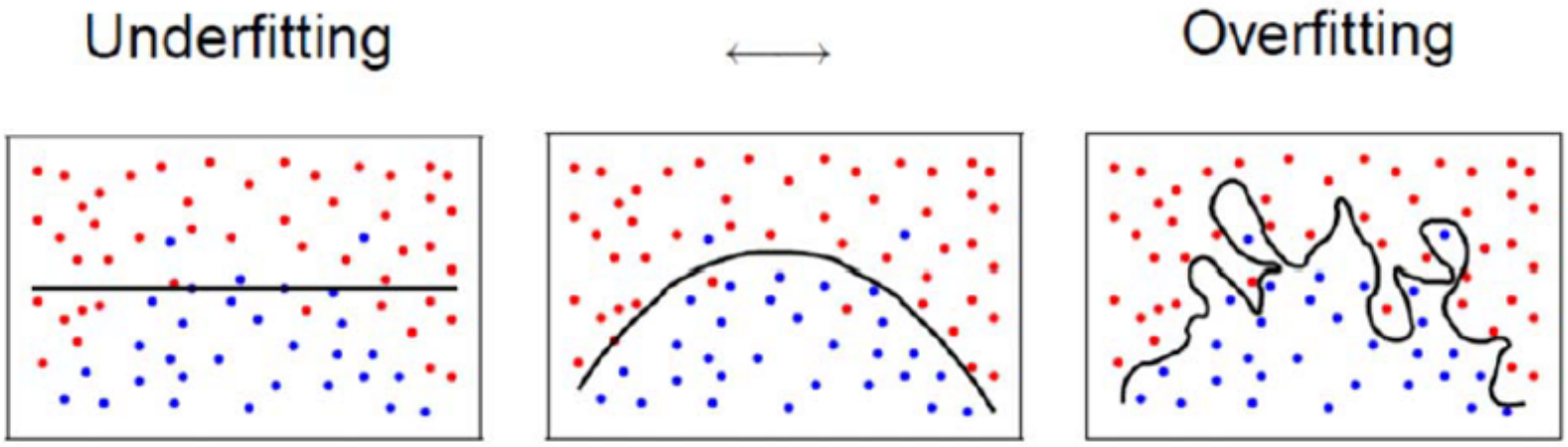
LINEAR SPLINE



HOW PICK # OF KNOTS?

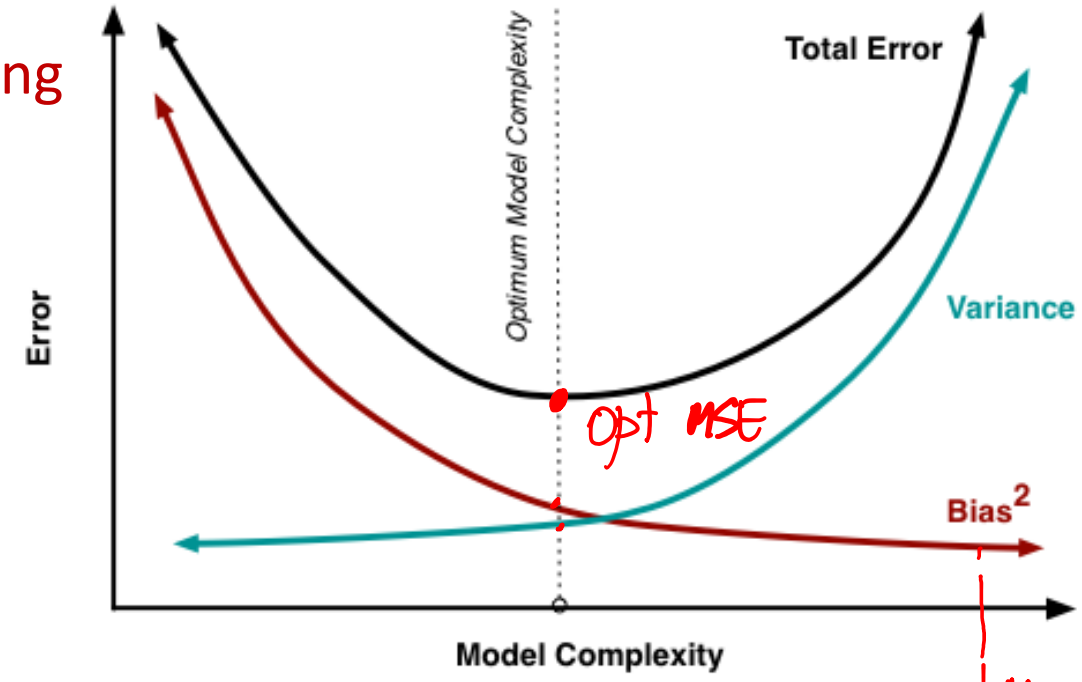
- Fit polynomial regression on each region (knot)
- Spline - Continuous and differentiable function at boundary

Generalization in ML



- Goal is to generalize well on new testing data
- Risk of overfitting to training data

Bias-Variance Tradeoff



- Bias = Difference between estimated and true models
- Variance = Model difference on different training sets

MSE is proportional to $\text{Bias}^2 + \text{Variance}$

Regularization

- A method for automatically controlling the complexity of the learned hypothesis
- **Idea**: penalize for large values of θ_j
  - Can incorporate into the cost function
  - Works well when we have a lot of features, each that contributes a bit to predicting the label
- Can also address overfitting by eliminating features (either manually or via model selection)

Reduce model complexity
Reduce model variance

Ridge regression

L_2 Regularization

$\|\theta\|_2^2$

- Linear regression objective function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{d} \theta_j^2$$

$\sim$ MSE

REGULARIZATION

λ - REG. PARAM

$\lambda = 0 \Rightarrow$ Linear Reg.

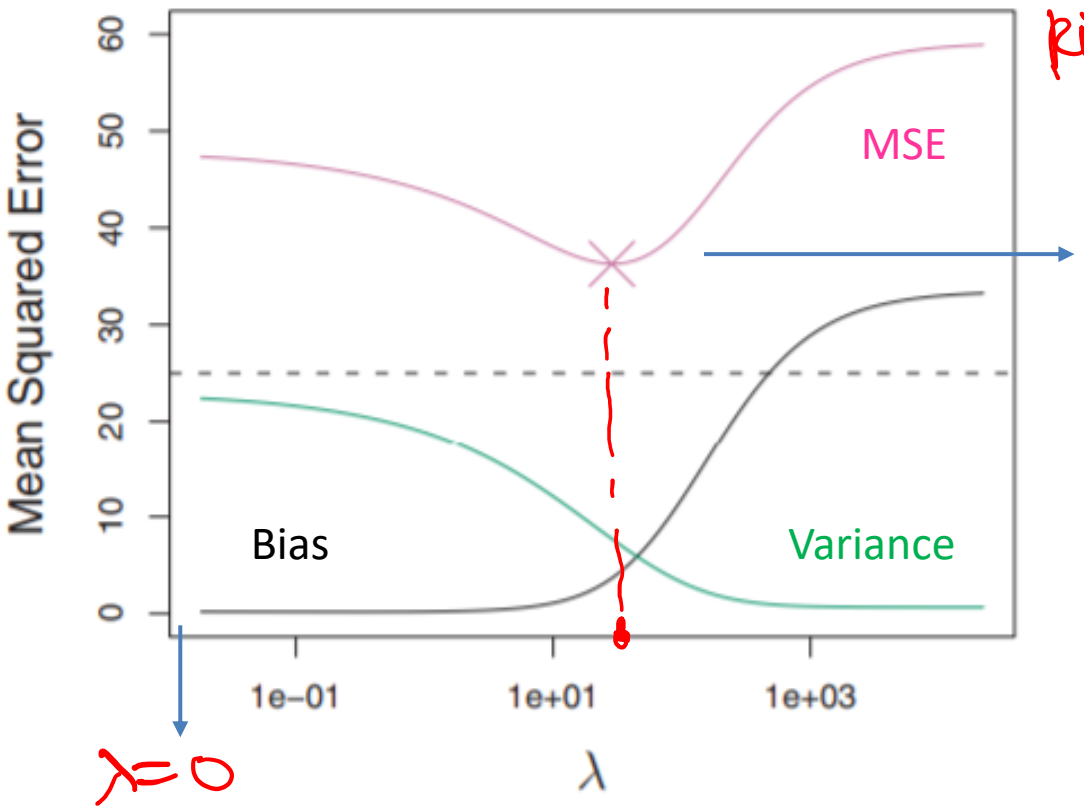
$\lambda \uparrow \Rightarrow \theta_j \downarrow$

DECREASE MODEL COMPLEXITY

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

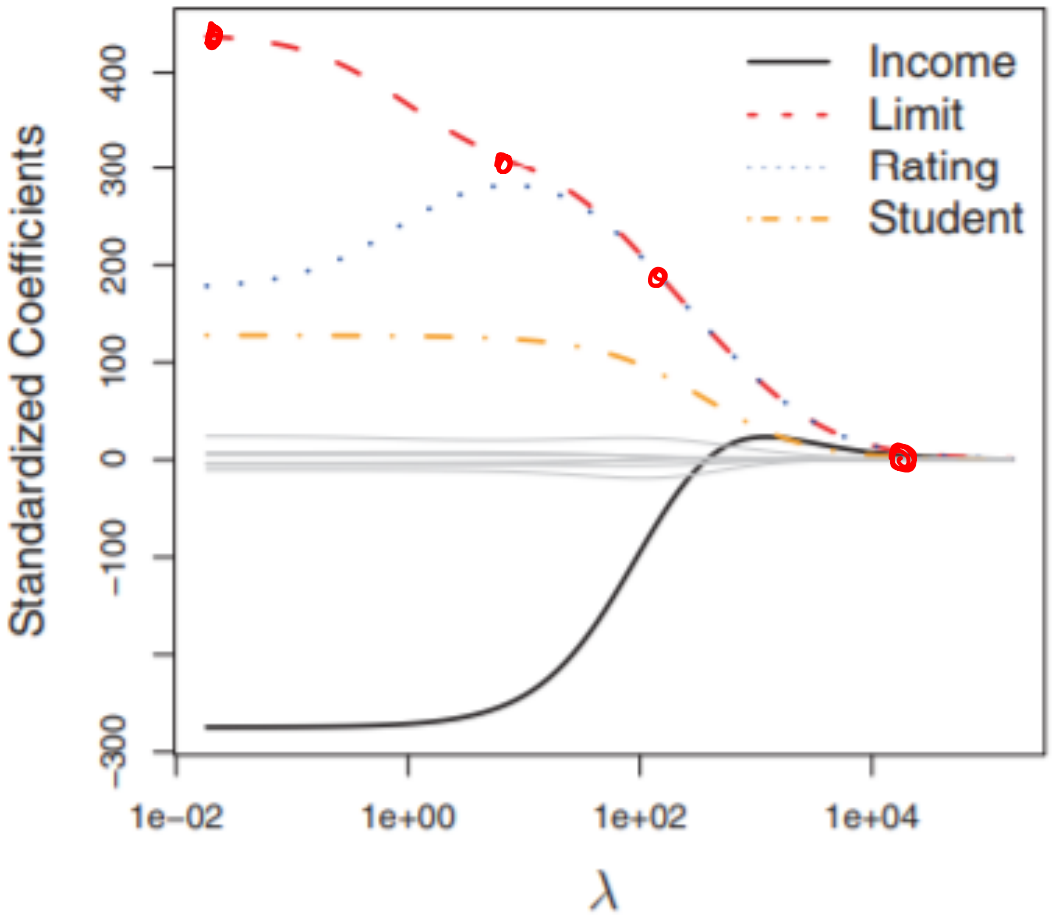
$J(\theta)$ convex $\Rightarrow$ HELPS GRADIENT DESCENT

Bias-Variance Tradeoff



Optimal
Ridge regression

Coefficient shrinkage



Predict credit card balance

GD for Ridge Regression

Min MSE

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{d} \theta_j^2$$

$\frac{\partial \theta_j^2}{\partial \theta_j} = 2\theta_j$

$\theta_1^2 + \theta_2^2 + .. + \theta_j^2 + .. + \theta_d^2$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \cdot x_{ij} + \lambda\theta_j$$

$$\theta_j \leftarrow \theta_j - \alpha \cdot \sum_{i=1}^{N} (h_\theta(x_i) - y_i)\, x_{ij} - \alpha\lambda\theta_j$$

$$= \theta_j(1 - \alpha\lambda) - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i)\, x_{ij}, \quad j = 1, \ldots, d$$

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

TRAINING EX

$$x_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{id} \end{bmatrix}$$

y_i

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

Lasso Regression

$$J(\theta) = \sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} |\theta_j|$$

~ MSE

REG
L1 norm $\|\theta\|_1$

FEATURE SELECTION METHOD

- L1 norm for regularization
- Results in sparse coefficients
- Small issue: gradients cannot be computed around 0
  - Can use sub-gradient at 0

Alternative Formulations

- Ridge
  - L2 Regularization
  - $\min_{\theta} \sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2$ subject to $\sum_{j=1}^{d} |\theta_j|^2 \leq \epsilon$

MSE

ball of radius ϵ

- Lasso
  - L1 regularization
  - $\min_{\theta} \sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2$ subject to $\sum_{j=1}^{d} |\theta_j| \leq \epsilon$

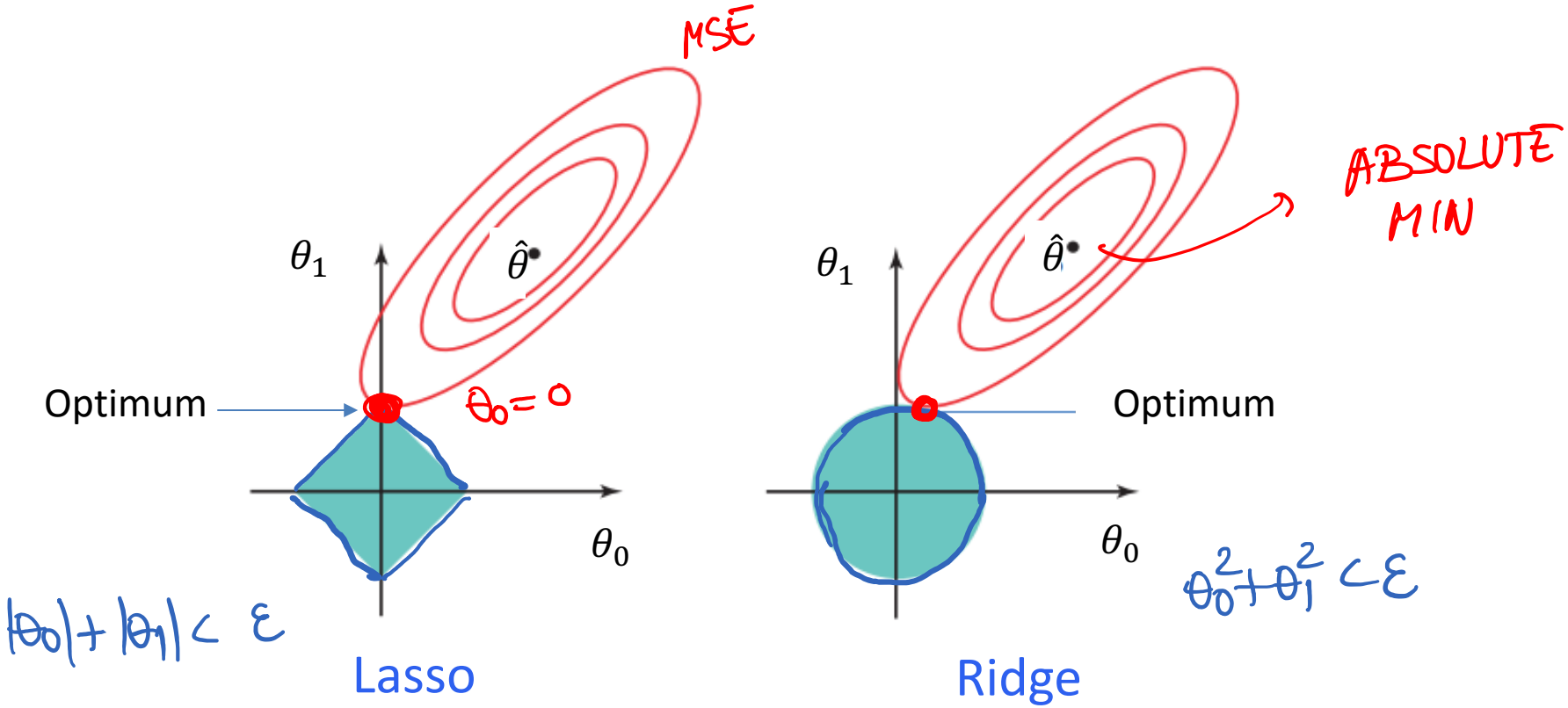
$|\theta_1| + |\theta_2| < \epsilon$

MSE

polytope

Lasso vs Ridge

- Ridge shrinks all coefficients
- Lasso sets some coefficients at 0 (sparse solution)
  - Perform feature selection



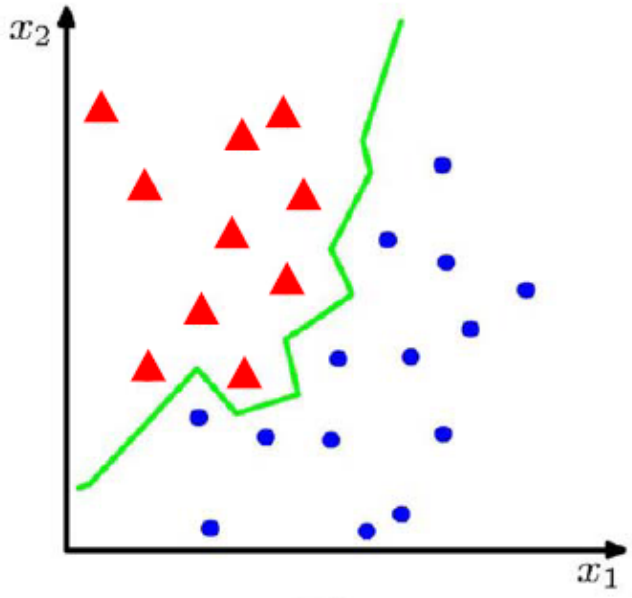
Ridge vs Lasso

- Both methods can be applied to any loss function MSE (regression or classification)

- Ridge
  - + CONVEX OBJECTIVE
  - + GLOBAL OPT
  - + CAN USE GRADIENT DESCENT
  - + HAS CLOSED FORM

- Lasso
  - + LARGE FEATURE SPACE (FEATURE SELECTION)
  - − NEED ADAPT GRADIENT DESCENT

Classification



Binary or discrete

- Suppose we are given a training set of N observations

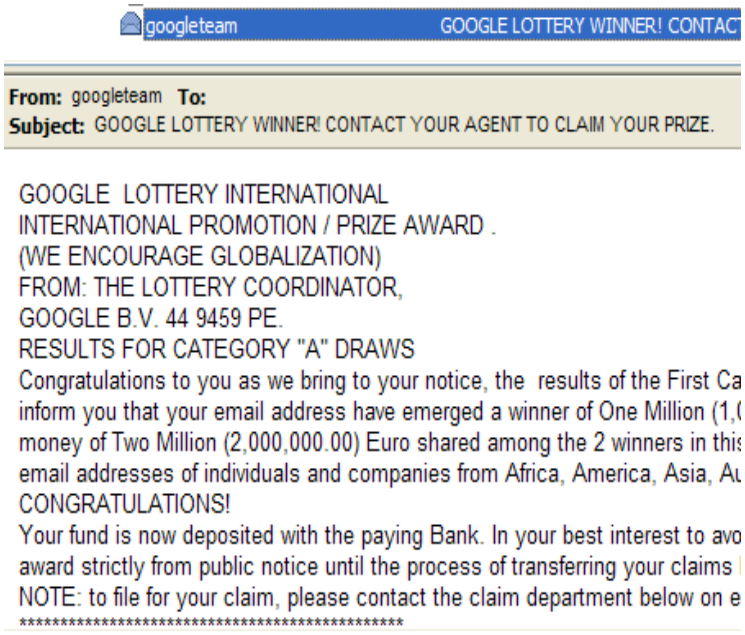
$$\{x_1, \ldots, x_N\} \text{ and } \{y_1, \ldots, y_N\}, x_i \in R^d, y_i \in \{0, 1\}$$

- Classification problem is to estimate f(x) from this data such that

$$f(x_i) = y_i$$

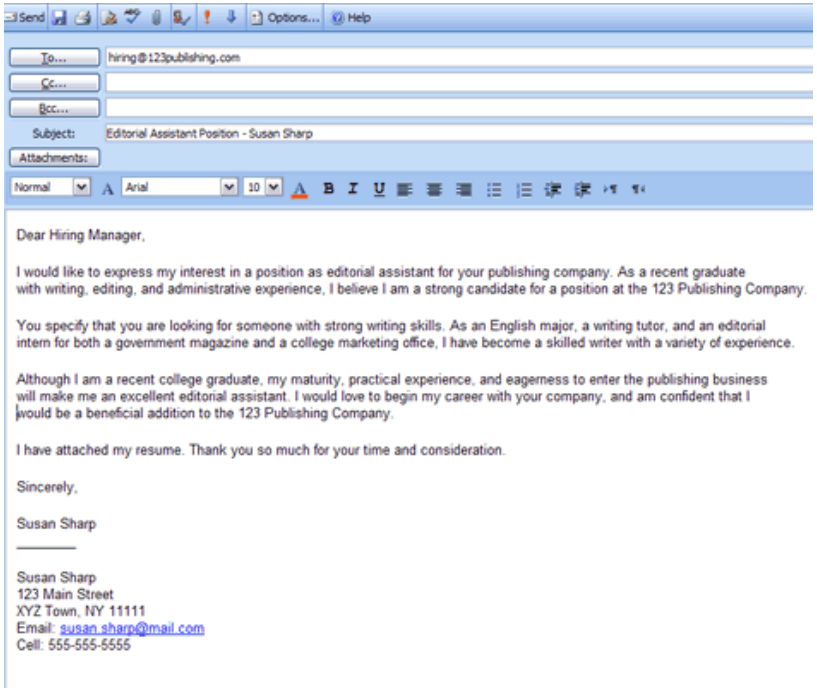
Example 1: Binary classification

Classifying spam email


googleteam GOOGLE LOTTERY WINNER! CONTACT

From: googleteam **To:**
Subject: GOOGLE LOTTERY WINNER! CONTACT YOUR AGENT TO CLAIM YOUR PRIZE.

GOOGLE LOTTERY INTERNATIONAL
INTERNATIONAL PROMOTION / PRIZE AWARD .
(WE ENCOURAGE GLOBALIZATION)
FROM: THE LOTTERY COORDINATOR,
GOOGLE B.V. 44 9459 PE.
RESULTS FOR CATEGORY "A" DRAWS
Congratulations to you as we bring to your notice, the results of the First Ca
inform you that your email address have emerged a winner of One Million (1,(
money of Two Million (2,000,000.00) Euro shared among the 2 winners in this
email addresses of individuals and companies from Africa, America, Asia, Au
CONGRATULATIONS!
Your fund is now deposited with the paying Bank. In your best interest to avo
award strictly from public notice until the process of transferring your claims
NOTE: to file for your claim, please contact the claim department below on e
****


Send Options... Help

To... hiring@123publishing.com
Cc...
Bcc...
Subject: Editorial Assistant Position - Susan Sharp
Attachments:

Normal Arial 10

Dear Hiring Manager,

I would like to express my interest in a position as editorial assistant for your publishing company. As a recent graduate with writing, editing, and administrative experience, I believe I am a strong candidate for a position at the 123 Publishing Company.

You specify that you are looking for someone with strong writing skills. As an English major, a writing tutor, and an editorial intern for both a government magazine and a college marketing office, I have become a skilled writer with a variety of experience.

Although I am a recent college graduate, my maturity, practical experience, and eagerness to enter the publishing business will make me an excellent editorial assistant. I would love to begin my career with your company, and am confident that I would be a beneficial addition to the 123 Publishing Company.

I have attached my resume. Thank you so much for your time and consideration.

Sincerely,

Susan Sharp

Susan Sharp
123 Main Street
XYZ Town, NY 11111
Email: susan.sharp@mail.com
Cell: 555-555-5555

Content-related features

- Use of certain words
- Word frequencies
- Language
- Sentence

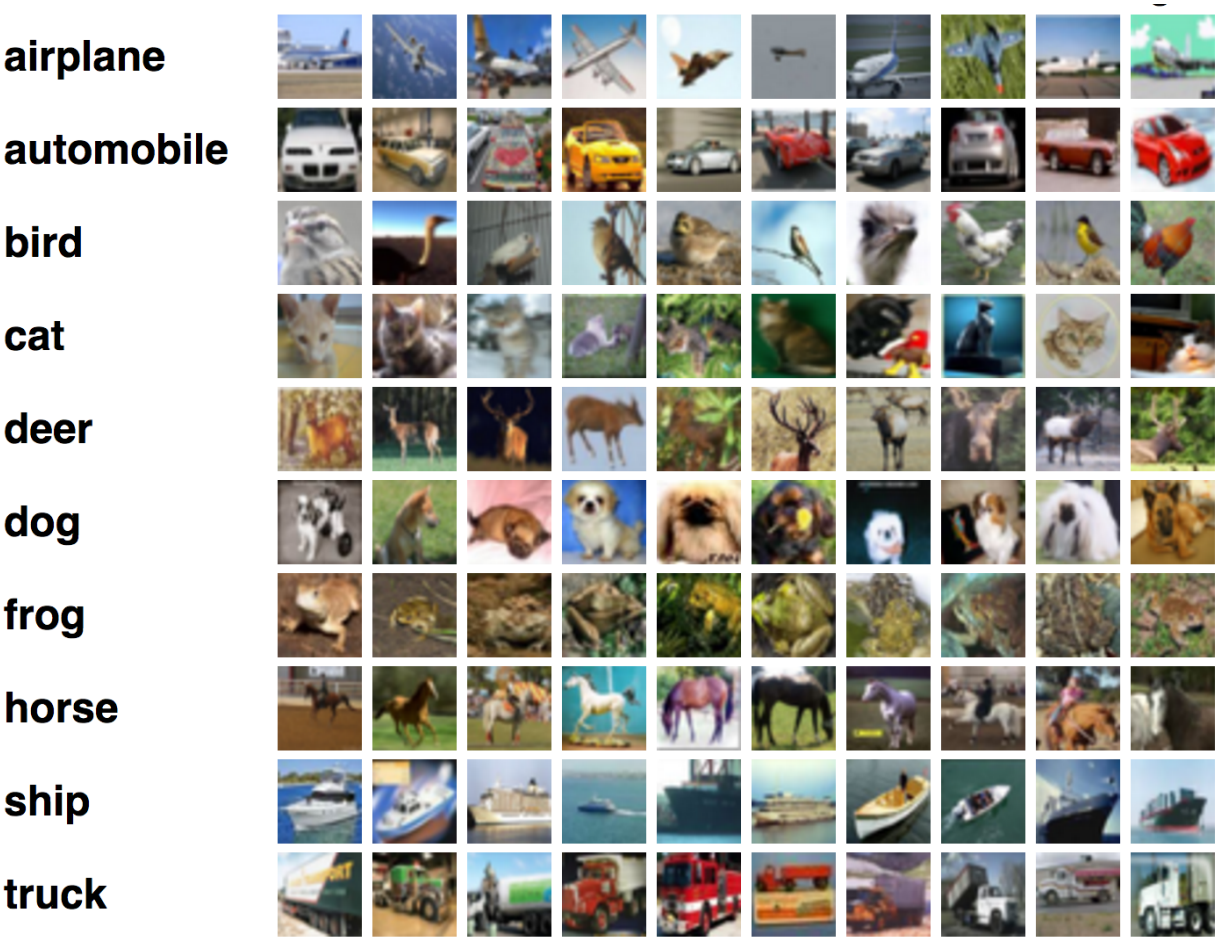
Structural features

- Sender IP address
- IP blacklist
- DNS information
- Email server
- URL links (non-matching)

Binary classification: SPAM or HAM

Example 2: Multi-class classification

Image classification



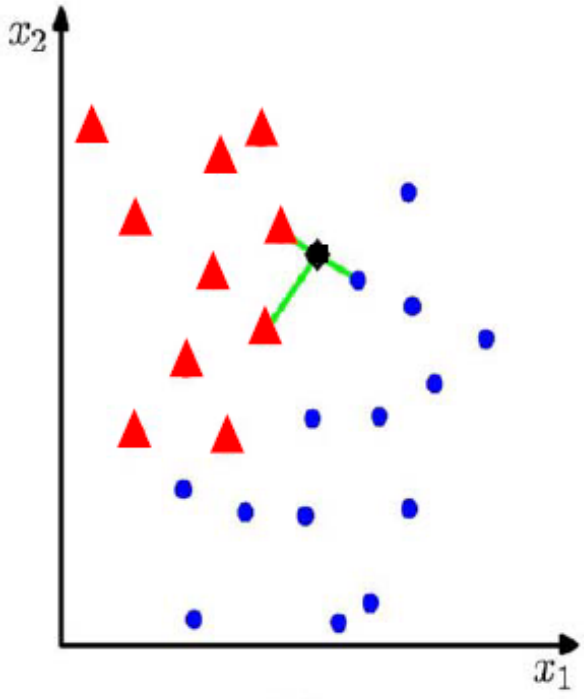
Multi-class classification

K Nearest Neighbour (K-NN) Classifier

Algorithm

- For each test point, x, to be classified, find the K nearest samples in the training data
- Classify the point, x, according to the majority vote of their class labels

e.g. K = 3

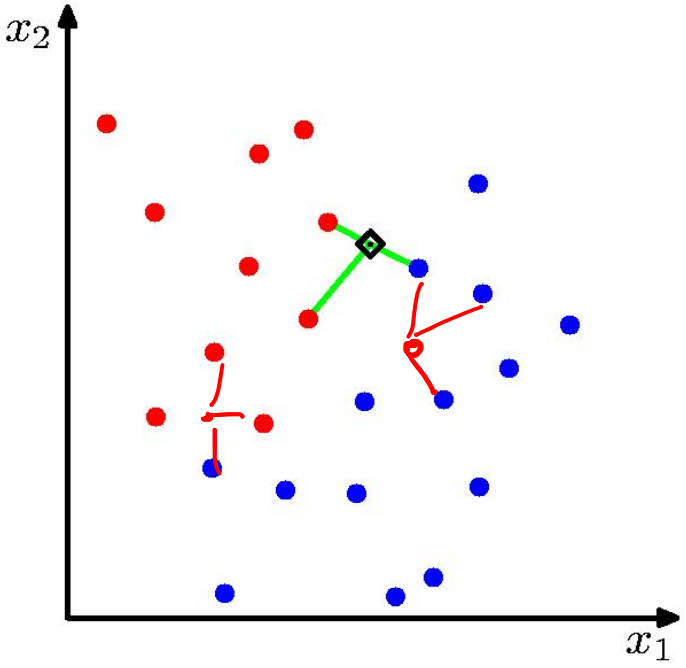


• applicable to multi-class case

Distance Metrics

- Euclidean Distance $\sqrt{(\sum_{i=1}^{k}(x_i - y_i)^2)}$ L_2
- Manhattan Distance $\sum_{i=1}^{k}|x_i - y_i|$ L_1
- Minkowski Distance $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$

kNN

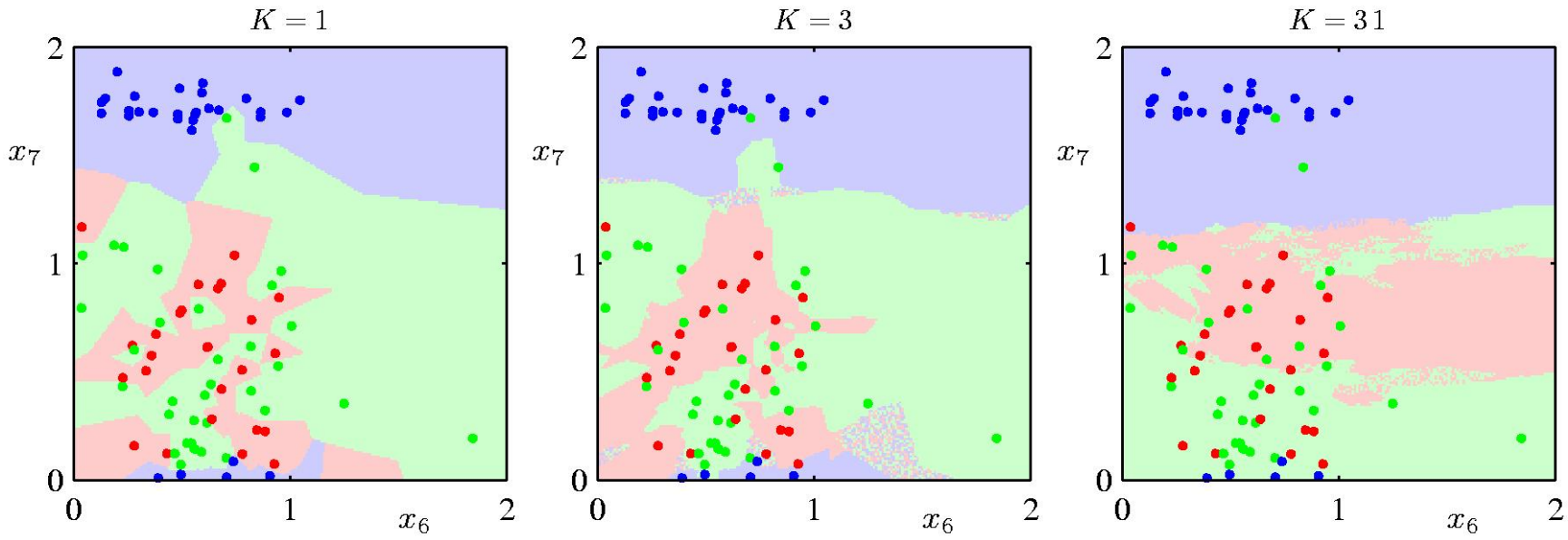


PARAM

- Algorithm (to classify point x)
  - Find k nearest points to x (according to distance metric)
  - Perform majority voting to predict class of x

ISSUES:
- DOES NOT LEARN
- INSTANCE LEARNER

K-Nearest-Neighbours for Multi-class Classification



Vote among multiple classes