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. 12, 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

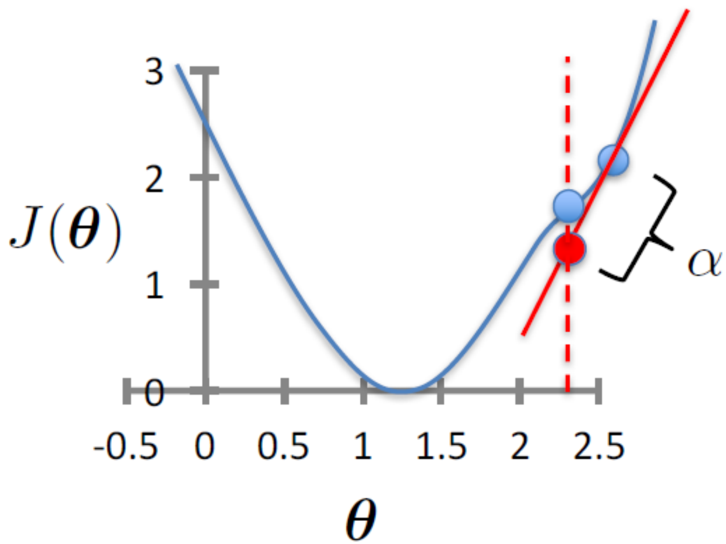
Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

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

simultaneous update
for j = 0 ... d

learning rate (small)
e.g., α = 0.05



- Gradient = slope of line tangent to curve
- Function decreases faster in negative direction of gradient
- Larger learning rate => larger step

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
  - $O(NTd)$, N is size of training data, d is feature dimension, and T is number of iterations
- 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	• Closed Form
+ Linear increase in d and N	+ No parameter tuning
+ Generally applicable	+ Gives the global optimum
- Need to choose α and stopping conditions	- Not generally applicable
- Might get stuck in local optima	- Slow: $O(Nd^2) + O(d^3)$

Issues with Gradient Descent

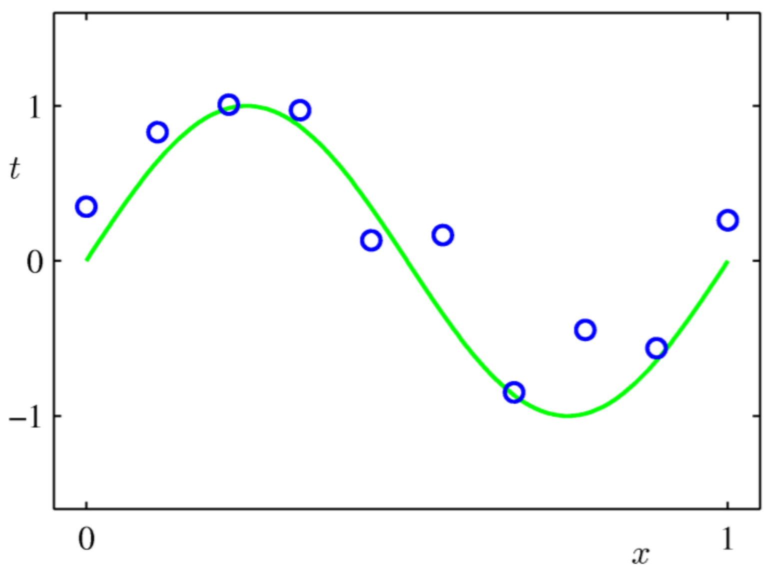
- Might get stuck in local optimum and not converge to global optimum
  - Restart from multiple initial points
- Only works with differentiable loss functions
- Small or large gradients
  - Feature scaling helps
- Tune learning rate
  - Can use line search for determining optimal learning rate

Review Gradient Descent

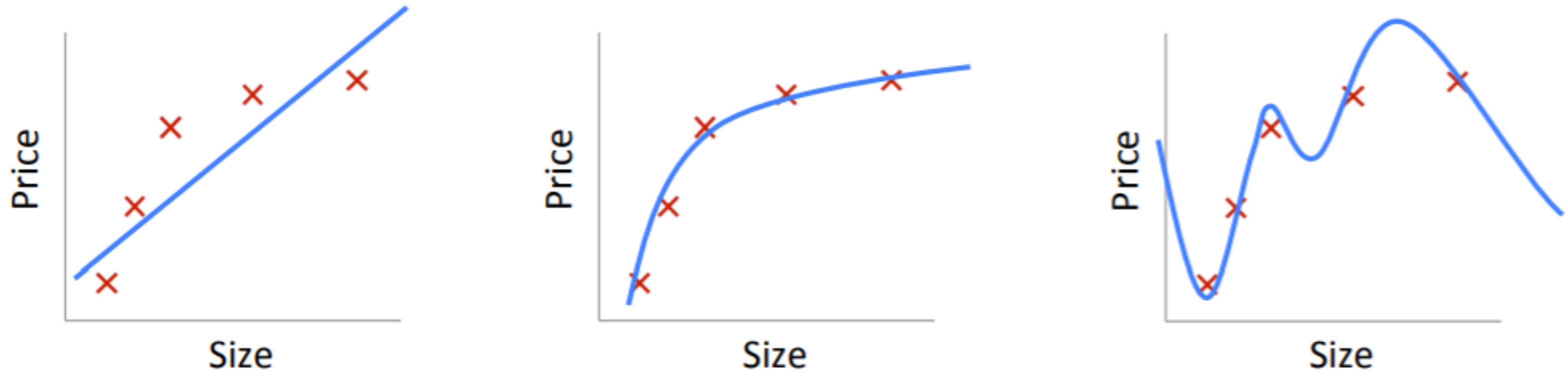
- Gradient descent is an efficient algorithm for optimization and training ML models
  - The most widely used algorithm in ML!
  - Much faster than using closed-form solution for linear regression
  - Main issues with Gradient Descent is convergence and getting stuck in local optima (for neural networks)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

Polynomial Regression

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



Polynomial Regression



$\theta_0 + \theta_1 x$

Underfitting (high bias)

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

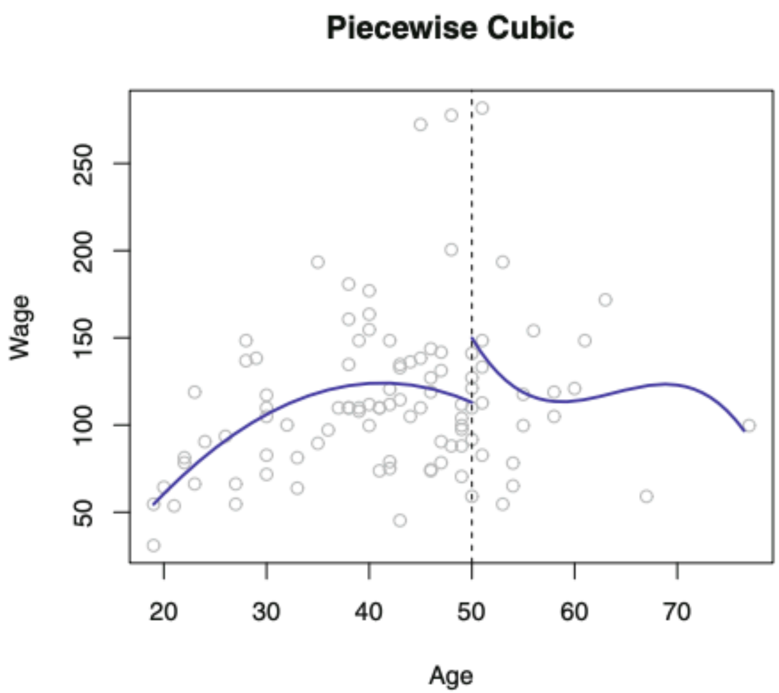
Correct fit

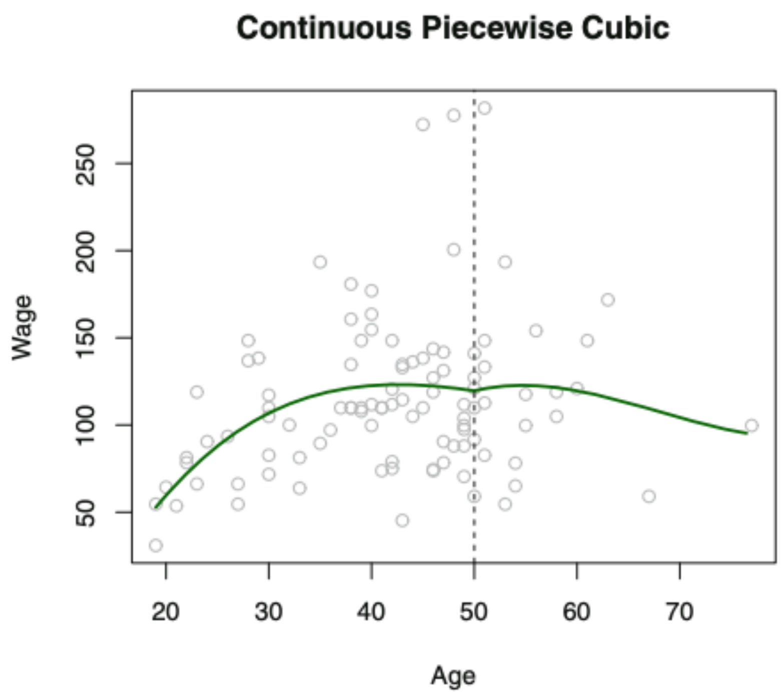
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfitting (high variance)

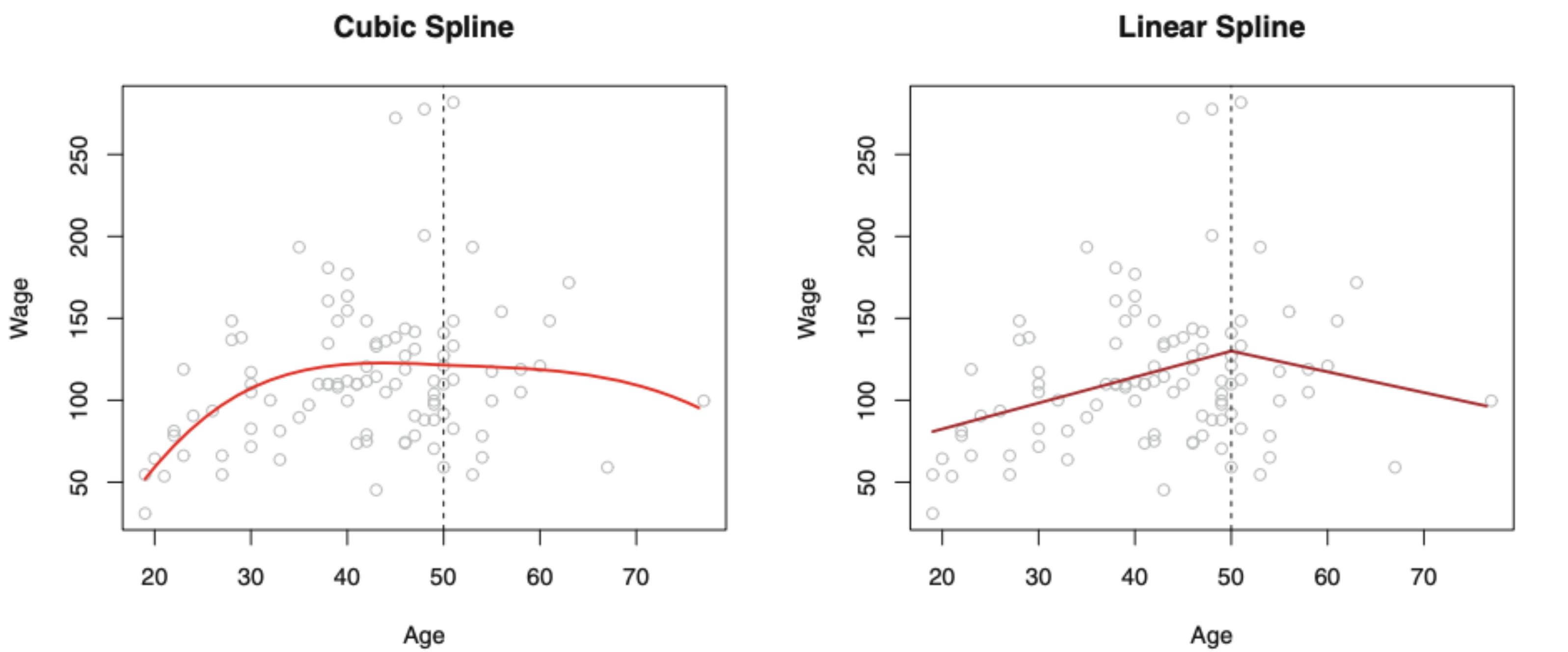
- Typically to avoid overfitting $p \leq 4$

Non-Linear Regression



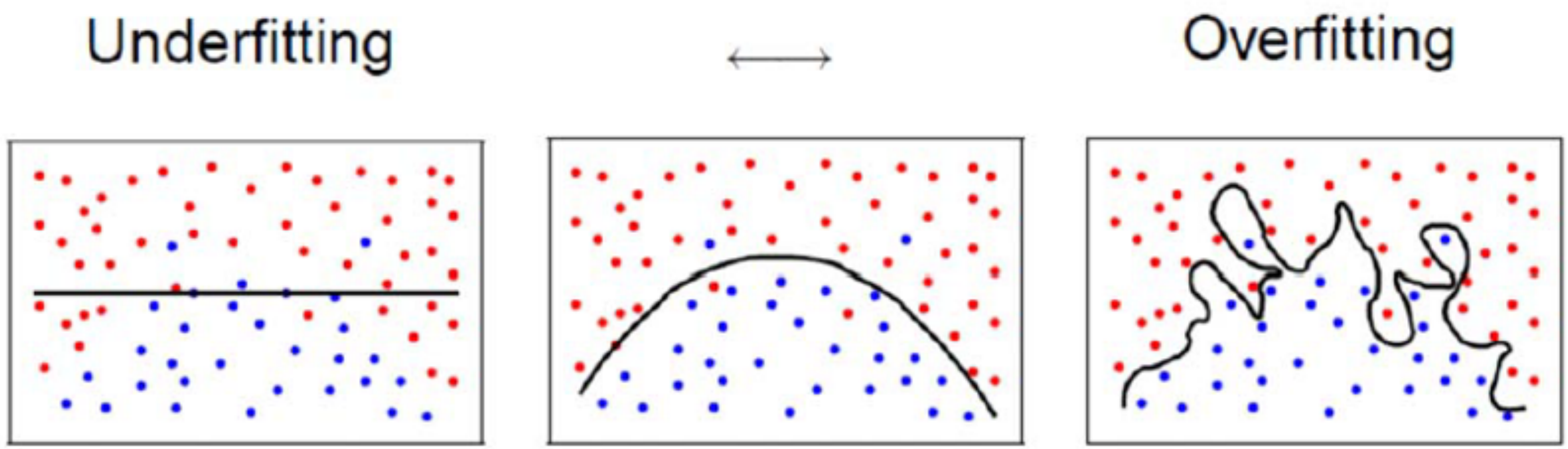


Spline Regression



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

Generalization in ML



Simple model

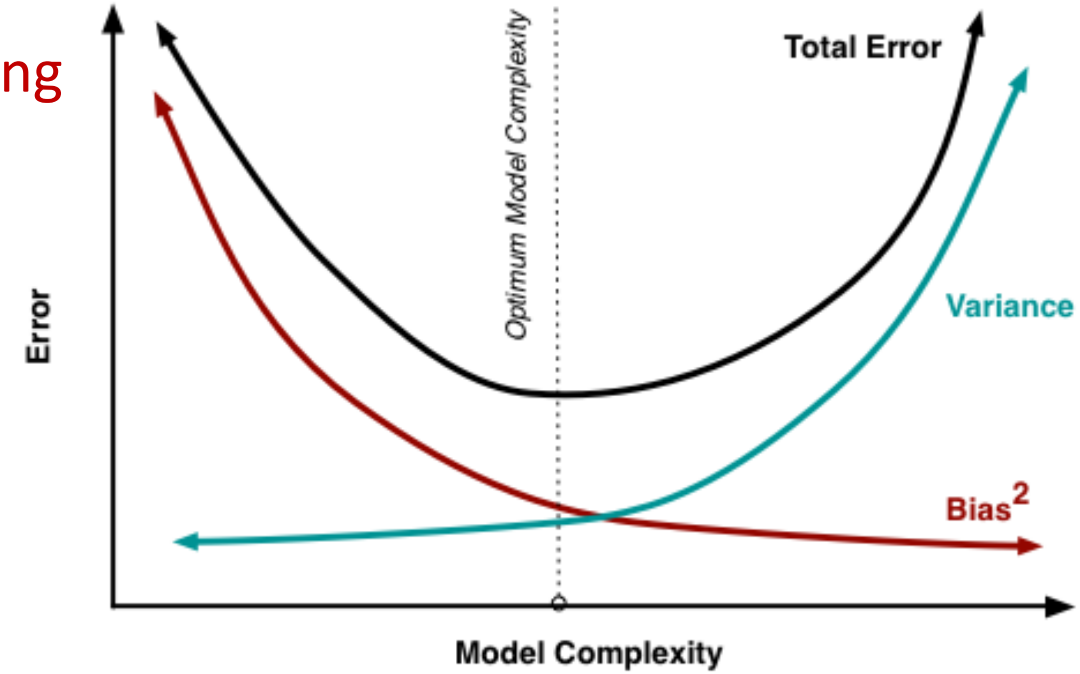
Complex model

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

Bias-Variance Tradeoff

Under-fitting

Over-fitting



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

MSE is proportional to Bias + 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

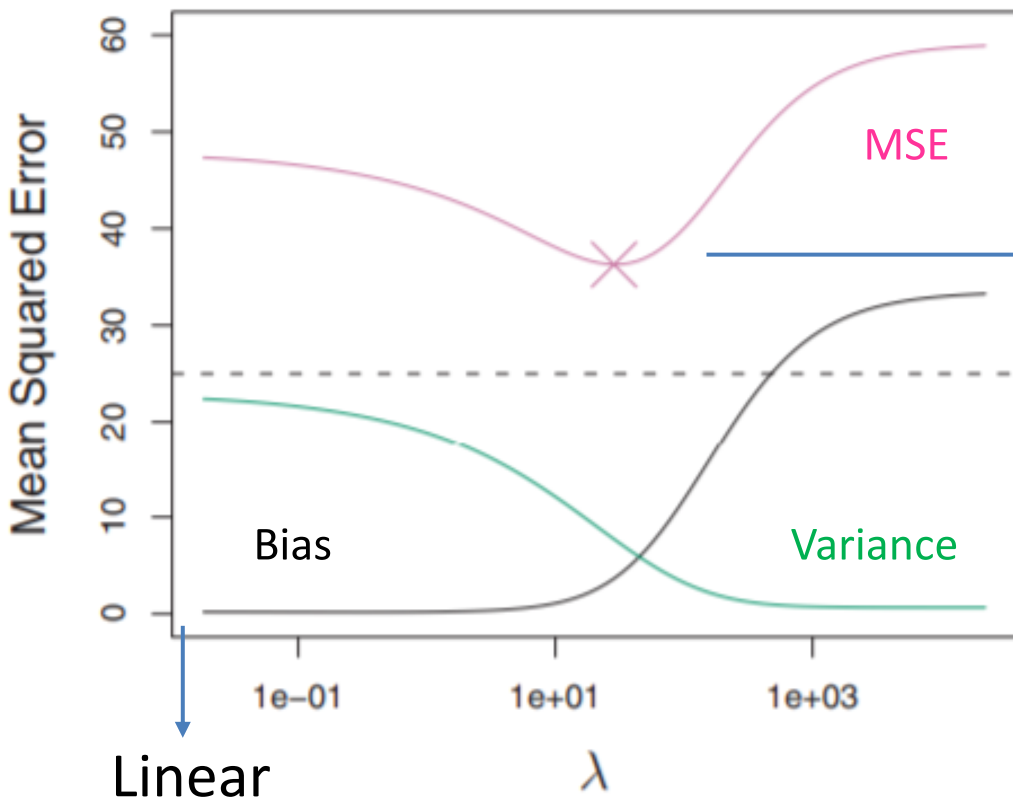
- Linear regression objective function

$$J(\theta) = \underbrace{\frac{1}{2}\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2}_{\text{model fit to data}} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{d} \theta_j^2}_{\text{regularization}}$$

  - λ is the regularization parameter ($\lambda \geq 0$)
  - No regularization on θ_0!

    - If λ = 0, we train linear regression
    - If λ is large, the coefficients will shrink close to 0

Bias-Variance Tradeoff

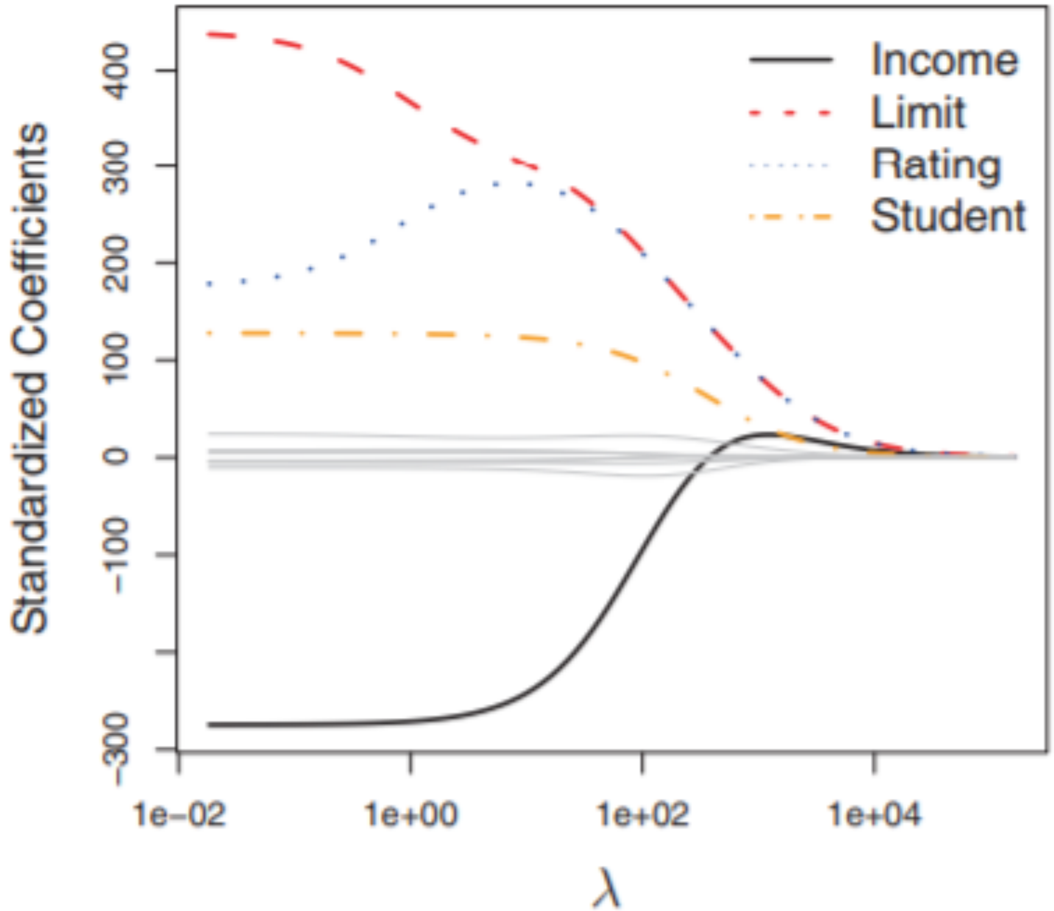


Optimal
Ridge regression

Linear
regression

Reduced model
complexity

Coefficient shrinkage



Predict credit card balance

GD for Ridge Regression

Min MSE

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

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

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N}(h_\theta(x_i) - y_i)x_{ij} - \underbrace{\alpha\lambda\theta_j}_{\text{Regularization}}$$

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

Lasso Regression

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

- L1 norm for regularization
- Results in sparse coefficients
- Issue: gradients cannot be computed around 0
- Method of sub-gradient optimization

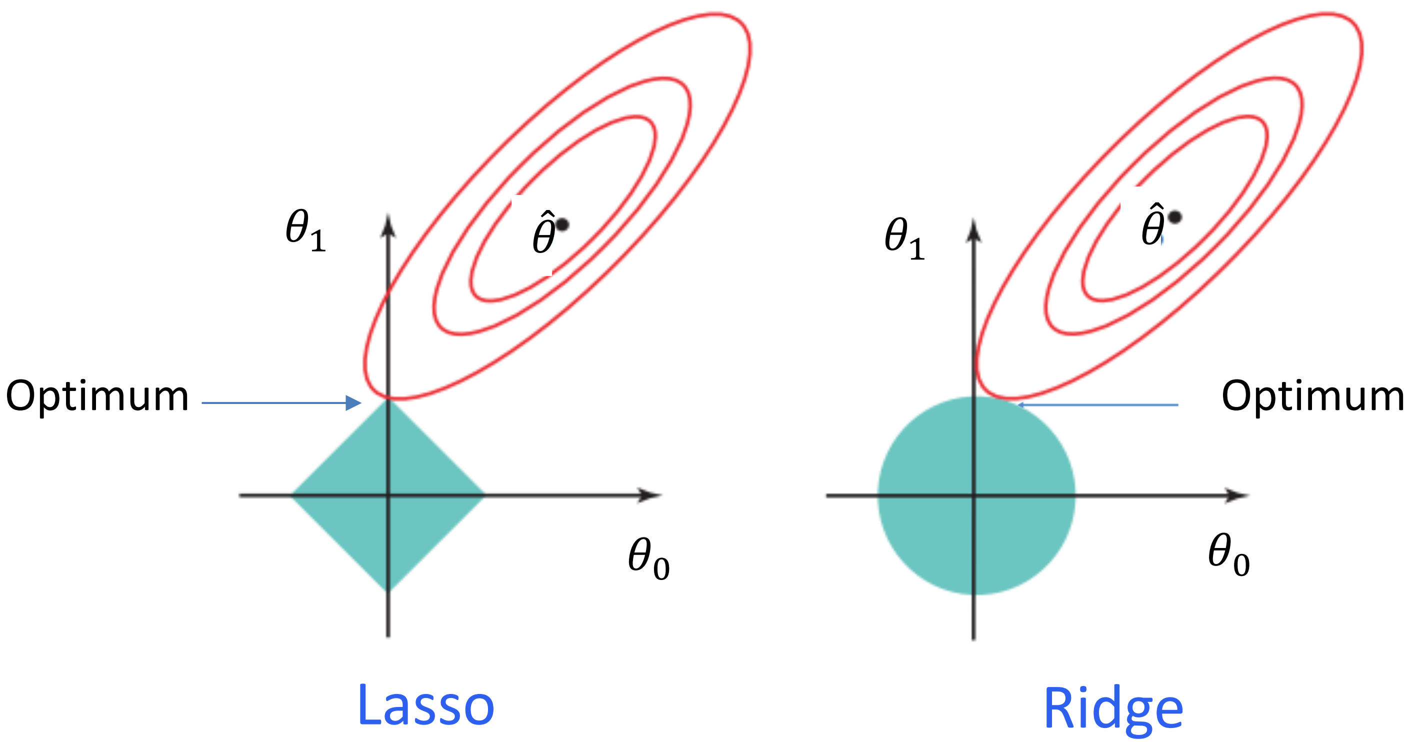
Alternative Formulations

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

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

Lasso vs Ridge

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

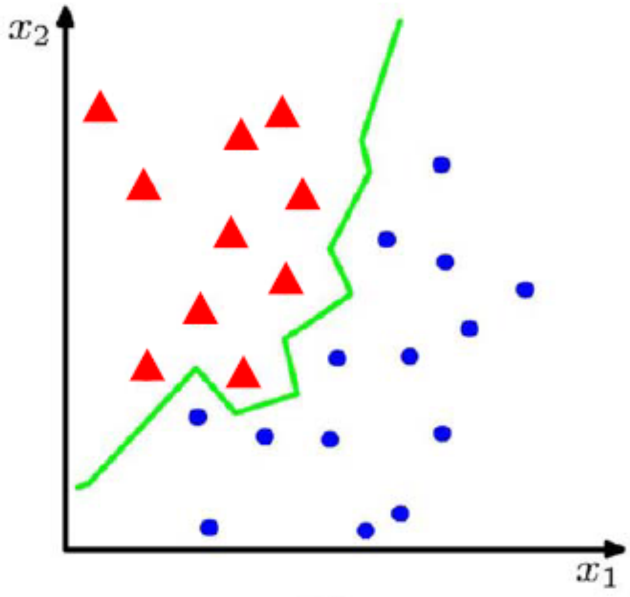


Ridge vs Lasso

- Both methods can be applied to any loss function (regression or classification)
- In both methods, value of regularization parameter λ needs to be adjusted
- Both reduce model complexity

Ridge	Lasso
+ Differentiable objective	- Gradient descent needs to be adapted
+ Gradient descent converges to global optimum	+ Results in sparse model
- Shrinks all coefficients	+ Can be used for feature selection in large dimensions

Classification



Binary or discrete

- Suppose we are given a training set of N observations

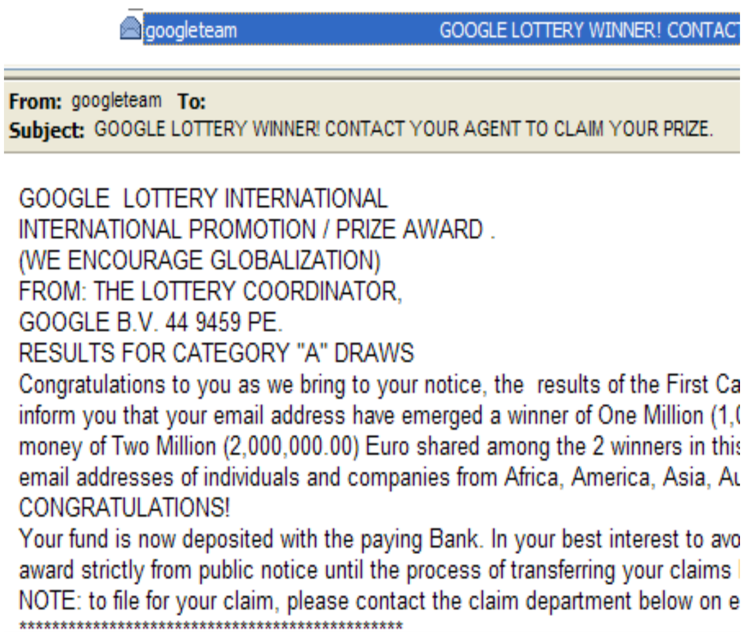
$$\{x_1, \dots, x_N\} \text{ and } \{y_1, \dots, y_N\}, x_i \in R^d, y_i \in \{0, 1\}$$

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

$$f(x_i) = y_i$$

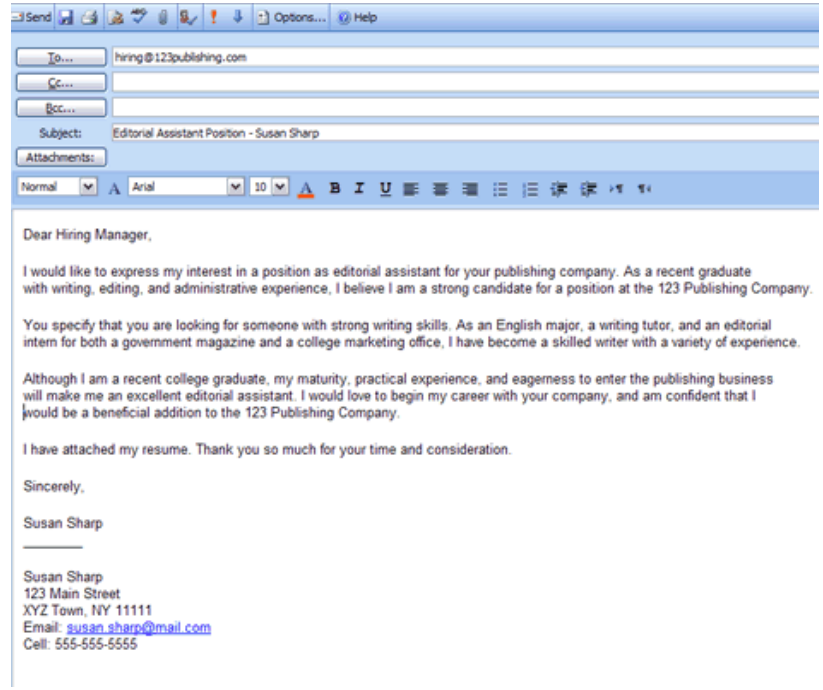
Example 1: Binary classification

Classifying spam email


googleteam GOOGLE LOTTERY WINNER! CONTAC

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:

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

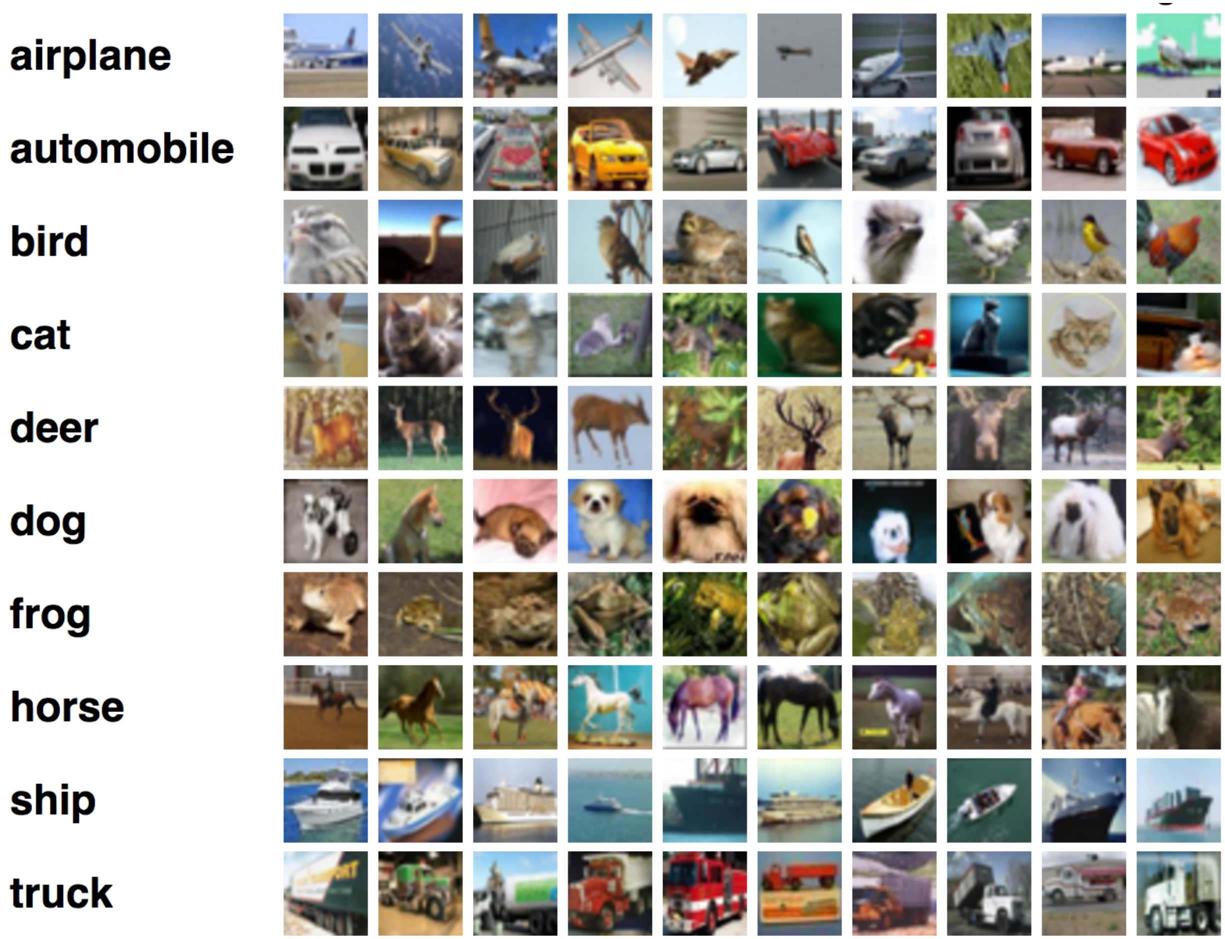
Structural features

- 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

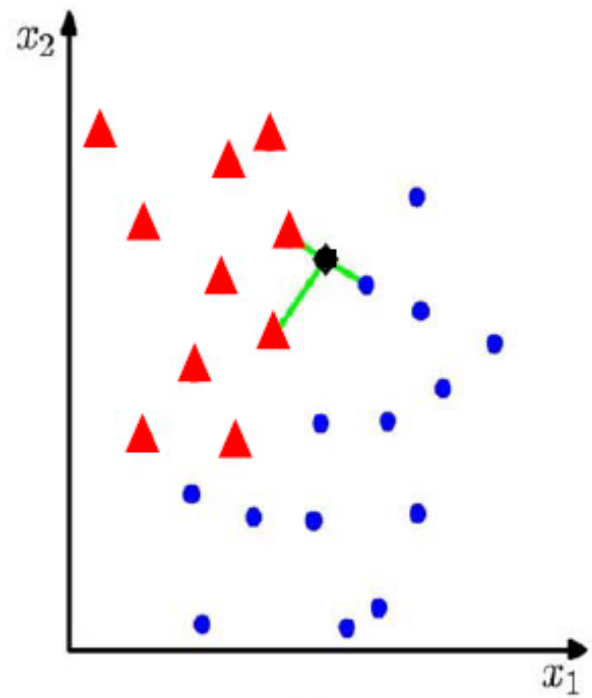
K Nearest Neighbour (K-NN) Classifier

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

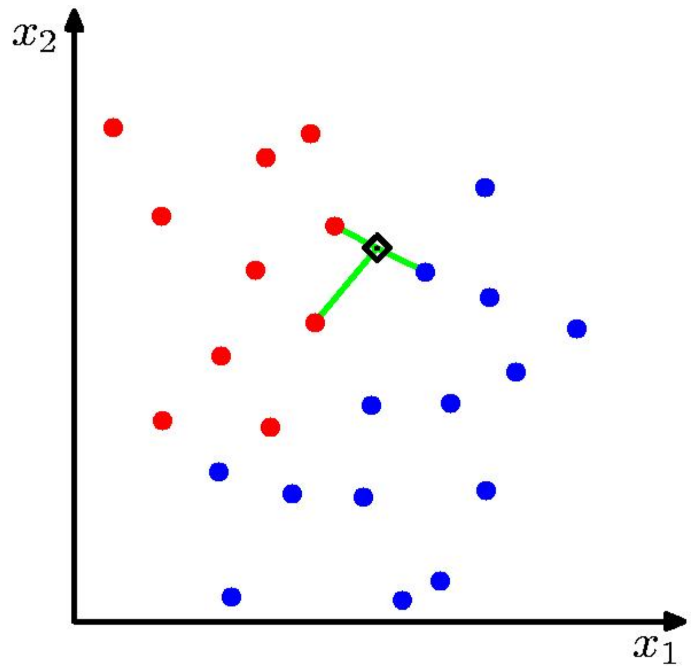
• applicable to multi-class case



Distance Metrics

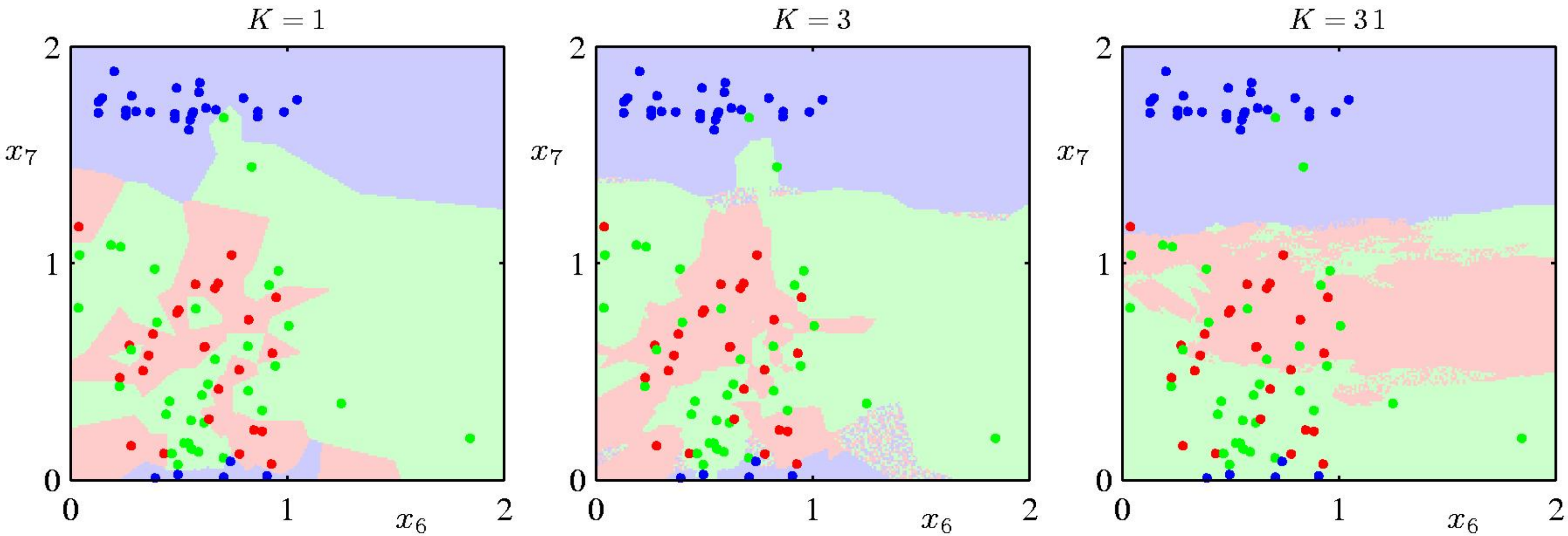
- Euclidean Distance $$\sqrt{(\sum_{i=1}^{k}(x_i - y_i)^2)}$$
- Manhattan Distance $$\sum_{i=1}^{k}|x_i - y_i|$$
- Minkowski Distance $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

kNN



- Algorithm (to classify point x)
  - Find k nearest points to x (according to distance metric)
  - Perform majority voting to predict class of x
- Properties
  - Does not learn any model in training!
  - Instance learner (needs all data at testing time)

K-Nearest-Neighbours for Multi-class Classification



Vote among multiple classes