Lecture 6: k-Nearest Neighbors

Instructor: Saravanan Thirumuruganathan

Outline

1. Introduction to Classification
2. k-NN (Nearest Neighbor) Classifier

In-Class Quizzes

- **URL:** `http://m.socrative.com/`
- **Room Name: 4f2bb99e**

Introduction to Classification

Major Tasks in Data Mining

- Predictive methods
  - Given some training data, build a model and use it to predict some variables of interest for unseen data
- Descriptive methods
  - Given some data, identify some significant, novel and useful patterns in the data that are interpretable by humans

Data Mining Tasks

- Classification, Regression: Predictive
- Clustering, Association Rule mining: Descriptive

Types of Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

- **Dataset:**
  - Training (labeled) data: $D = \{(x_i, y_i)\}$
  - $x_i \in \mathbb{R}^d$
  - Test (unlabeled) data: $x_0 \in \mathbb{R}^d$
- **Tasks:**
  - Classification: $y_i \in \{1, 2, \ldots, C\}$
  - Regression: $y_i \in \mathbb{R}$
- **Objective:** Given x_0, predict y_0
- **Supervised** learning as y_i was given during training

Unsupervised Learning

- Given: dataset $D = \{x_i\}$
- Objective: Find interesting patterns without explicit supervision
- **Tasks:**
  - Clustering
  - Outlier detection
  - Dimensionality reduction
  - Many more

Reinforcement Learning

- Training "agents" to take actions to maximize rewards
- Reinforcement is given via action-reward
- Objective: Find out what is the optimal action $\mathbf{a}$ to take when in state $\mathbf{x}$, in order to maximize long-term reward
- **Examples:** Learning correct answers from score, self-driving cars, learning to fly helicopters autonomously, learning to play games

Classification Methods

- **Model based:** Build a (simple) model from the training data and use it to predict unseen data
- **Memory based:** Keep in memory all training data and use it to predict unseen data

Classification Models

Some of the methods we will discuss in the class:

- Tree based: Decision and Regression trees
- Instance based: Nearest Neighbor
- Bayesian and Naive Bayes
- Neural Networks and Deep Learning
- Support Vector Machines

Binary and Multi-Class Classification

- $C = 2$: Predict which of the two classes for the unseen record
  - Spam or Ham for emails
  - Benign or malignant for tumours
- $C > 2$: Multi-Class classification - predict the right class.
  - Categorize mail as important, social, unimportant
  - Identify color of eyes
  - Identify wine type from features
- Multi-class classification is often much more harder

Trade-offs

- Prediction accuracy versus interpretability
- Good fit versus over-fit or under-fit
- Parsimony versus black-box

Classification Design Cycle[1]

1. Collect data and labels (the real effort)
2. Choose features (the real ingenuity)
3. Pick a classifier (some ingenuity)
4. Train the classifier (some knobs, fairly mechanical)
5. Evaluate the classifier (needs care)

[1] http://www.cs.sun.ac.za/~kroon/courses/machine_learning/lecture2/kNN-intro_to_ML.pdf

k-NN Classifier

Instance based Classifiers

- Store ALL the training data
- Use the training data to predict class label for a new record
- Common Examples:
  - Rote-Learner: Memorize entire training data, predict value if the new record matches some training data
  - Nearest Neighbor: Use k points closest to new record to perform classification

Nearest Neighbor Methods

- Non-parametric, model-free approaches
- Formalized in 1960s
- Simple to understand and implement

Why k-NN

- One of the top-10 Data Mining algorithms[2]
- 1-**NN Error bounds:**
  - When number of training data n tends to ∞ in a C-Class problem then the 1-NN error rate (1NNER) is bounded by

$$BER \leq 1NNER \leq BER \times \left(2 - \frac{C}{C-1} \times BER\right)$$

  - 1-NN Error rate is at most twice that of BER
- Asymptotically Consistent: With infinite training data and large enough k, k-NN approaches the best possible classifier (Bayes Optimal)

[2] http://www.cs.umd.edu/~samir/498/10Algorithms-08.pdf

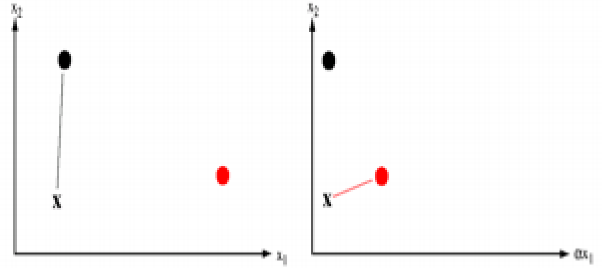
k-Nearest Neighbor

- Distance Metric: To compute the similarities between records
- k: How many neighbors to look at?
- A weighting function (optional)
- Decision strategy: Often simple majority voting

k-NN Algorithm

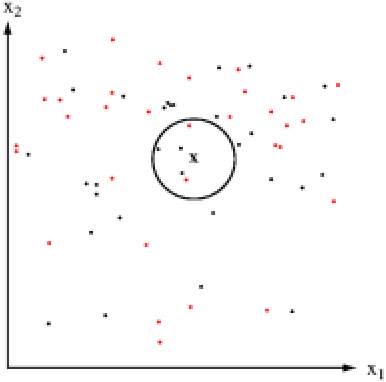
1. Compute the test point's distance from each training point
2. Sort the distances in ascending (or descending) order
3. Use the sorted distances to select the k nearest neighbors
4. Use majority rule (for classification) or averaging (for regression)

1-NN Example[3]



[3] http://www.lkozma.net/knn2.pdf

k-NN Example[4]

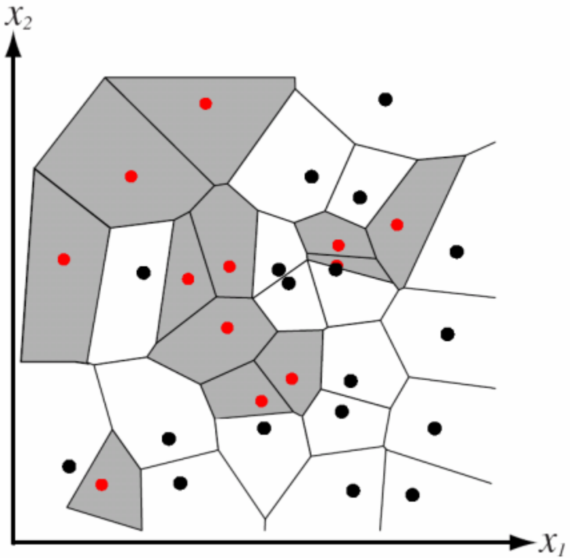


[4] http://www.lkozma.net/knn2.pdf

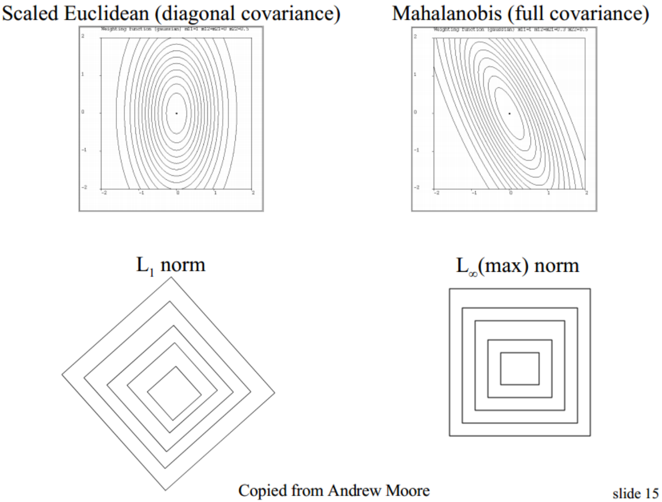
Distance Metric

- Used to compute similarity between entities
- If all values are numeric, Euclidean measure is often used

Voronoi Cells in 2D[5]



Distance Metric



Feature Normalization

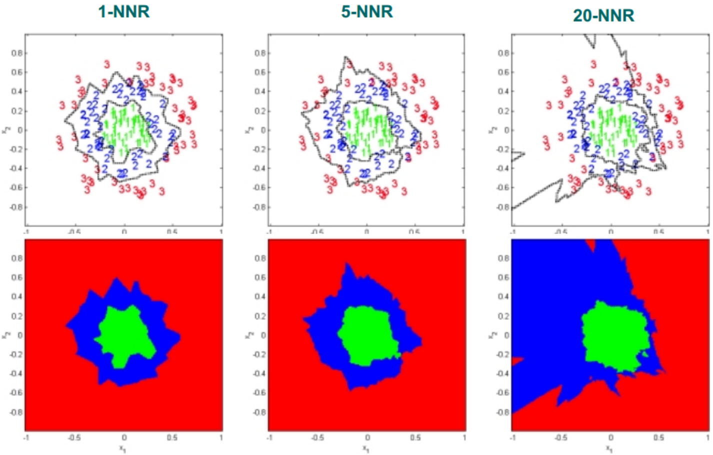
- Features should be on the **same** scale
- Example: if one feature has its values in millimeters and another has in centimeters, we would need to normalize
- Common way: Center and Normalize to get 0 mean and unit variance

$$z_i = \frac{x_i - \overline{x_i}}{\sigma}$$

Finding Optimal k

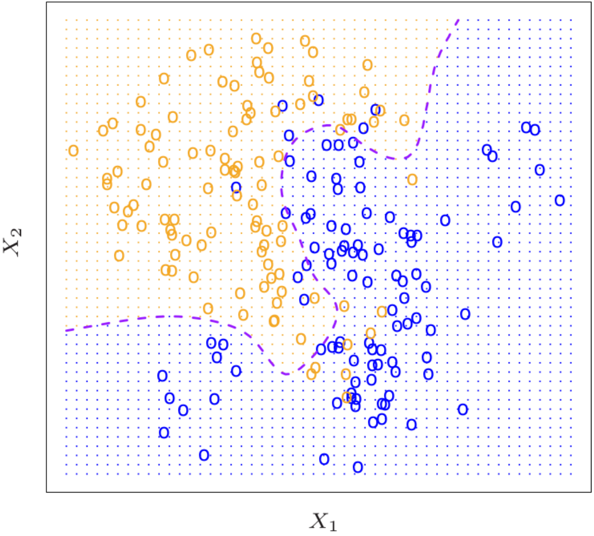
- Often k-NN has lower error rate than 1-NN
- But the error does not monotonically decrease
- Picking k: Cross validation

Impact of k[6]



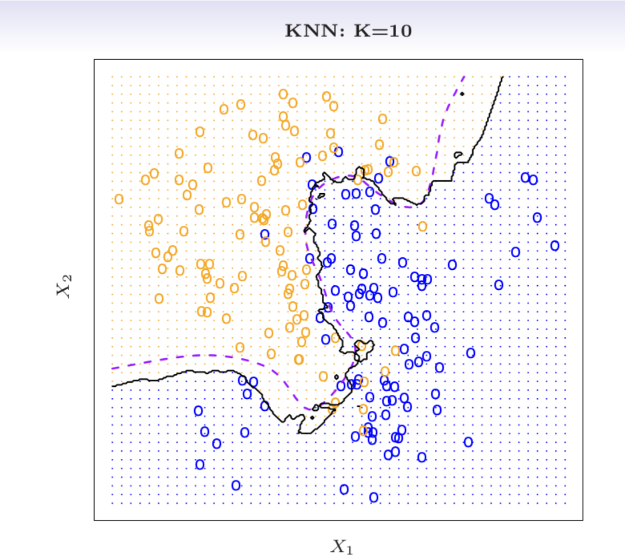
[6] http://courses.cs.tamu.edu/rgutier/cs790_w02/l8.pdf

Impact of k[7]

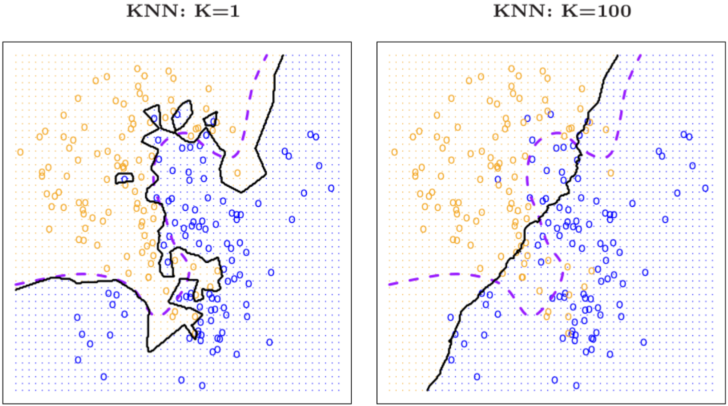


[7] ISLR

Impact of k^8



Impact of k[9]



[9] ISLR

Impact of k[10]

- Small k
  - Creates many small regions for each class
  - May lead to non-smooth decision boundaries and overfit
  - Leads to higher variance (i.e. classifier is less stable)
- Large k
  - Creates fewer larger regions
  - Usually leads to smoother decision boundaries (although, too smooth boundaries might underfit)
  - Leads to higher bias (i.e. classifier is less precise)

[10] http://www.cs.cornell.edu/courses/CS4758/2013sp/materials/cs4758-knn-lectureslides.pdf

Weighted k-NN

- Often you might want to use some weights
- Typically to give higher weights to points nearby than to points that are farther
- One possibility: $\frac{1}{dist^2}$ (i.e. inverse of squared distance)
- Alternatively, give more weight to similarity on important features

$$dist(x_i, x_j) = \sum_{k=1}^{d} w_k \, dist(x_{ik}, x_{jk})$$

Computational Complexity

- $O(nd)$ where n is training set size and d is the number of dimensions
- VERY expensive, computationally
- Often, special data structures such as Voronoi diagrams, KD-trees are used to speed things up.

Other Things to Watch Out

- Missing data (features) will cause problems
- Sensitive to class outliers
- Sensitive to irrelevant features (so ensure feature engineering and normalization are done first)

Summary

Major Concepts:

- Major data mining tasks
- Classification basics
- k-NN, variants - pros and cons