



# K Nearest Neighbors and Kernel Regression

GU 4241/GR 5241

Statistical Machine Learning

Xiaofei Shi



# Tasks

Input → Regressor → Predict real number

Input → Classifier → Predict category

Input → Density Estimator → Probability





# Types of classifiers

- Discriminative classifiers:
  - Directly estimate a decision rule/boundary
  - e.g. decision tree, SVM
- Instance based classifiers:
  - Use observation directly
  - e.g. K nearest neighborhood
- Generative classifiers:
  - Build a generative statistical model
  - e.g. Bayesian Network



# Loss function

Recall the loss function in previous classifiers:

- SVM: maximum margin + penalty
- Decision Tree: maximum information gain
- What else?



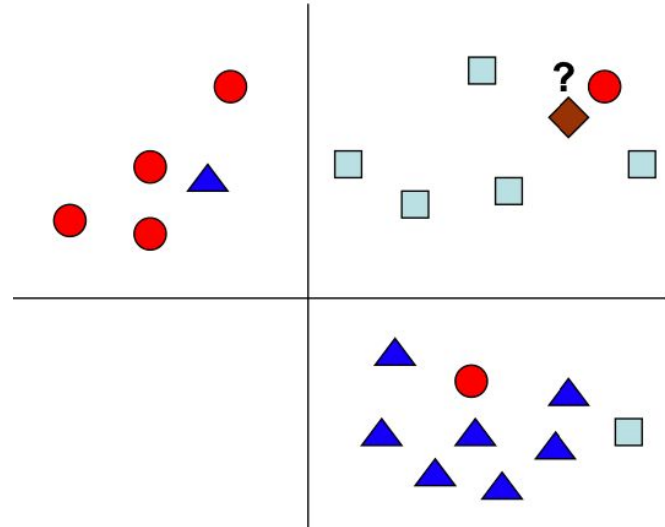
# Loss function

- The risk value we computed assumes that both errors (assigning instances of class 1 to class 0 and vice versa) are equally harmful.
- However, this is not always the case.
- Why?
- In general our goal is to minimize loss, often defined by a loss function:  $L_{0,1}(x)$  which is the penalty we pay when assigning instances of class 0 to class 1

$$E[L] = L_{0,1}p(y=0) \int_{L_1} p_0(x)dx + L_{1,0}p(y=1) \int_{L_0} p_1(x)dx$$

# K nearest neighbors (KNN)

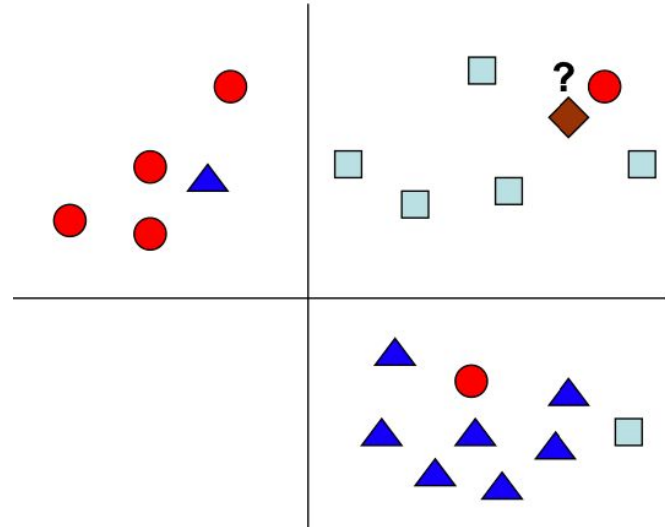
- A simple yet surprisingly efficient algorithm
- Requires the definition of a similarity measure or a distance function between sample points
- Select the class based on the majority vote among the K nearest sample points



# K nearest neighbors (KNN)

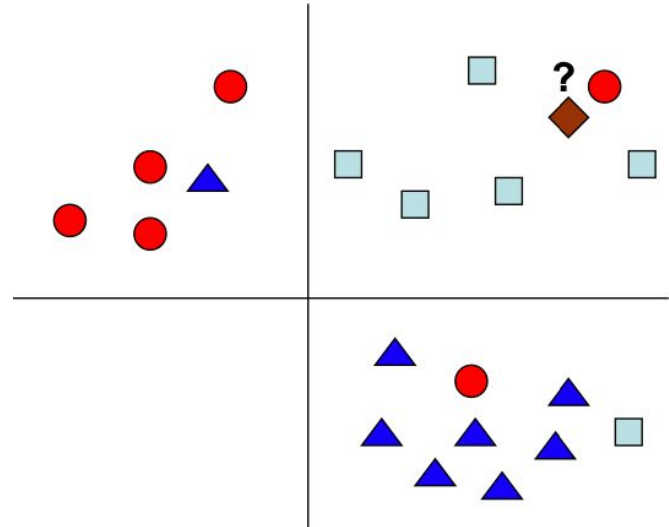
- A simple yet surprisingly efficient algorithm
- Requires the definition of a similarity measure or a distance function between sample points
- Select the class based on the majority vote among the K nearest sample points

What is the best value of K?



# K nearest neighbors (KNN)

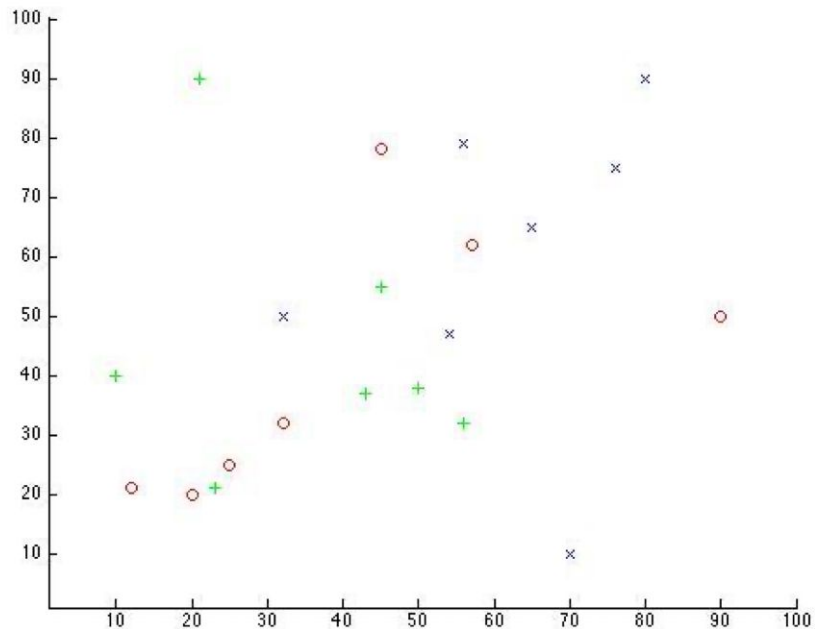
- Choice of  $K$  influences the “smoothness” of the resulting classifier
- In this sense, the KNN classifier is similar to a kernel methods
- However, the smoothness of the classifier should be determined by the actual distribution of the data, i.e. the density function  $p(x)$  of the data, not any predefined parameter.



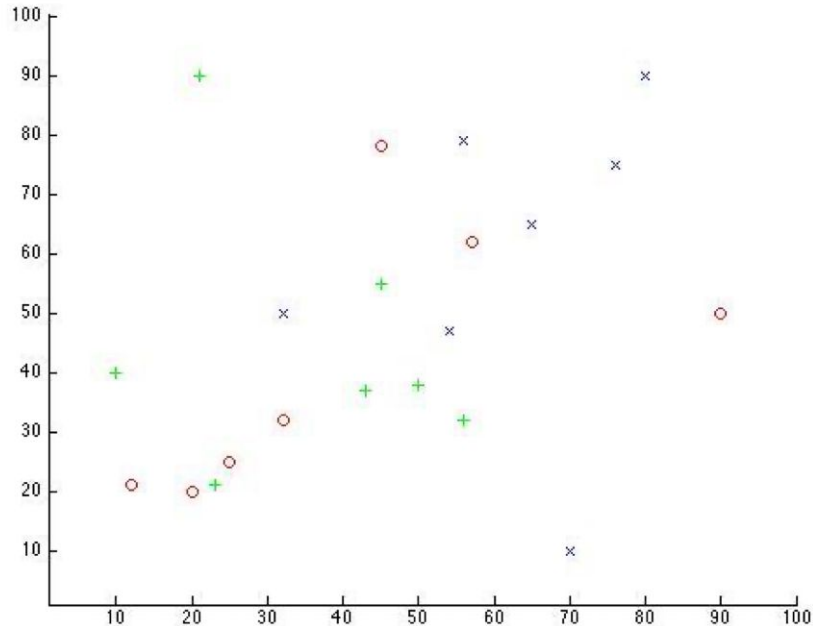




# The effect of increasing K



# The effect of increasing K

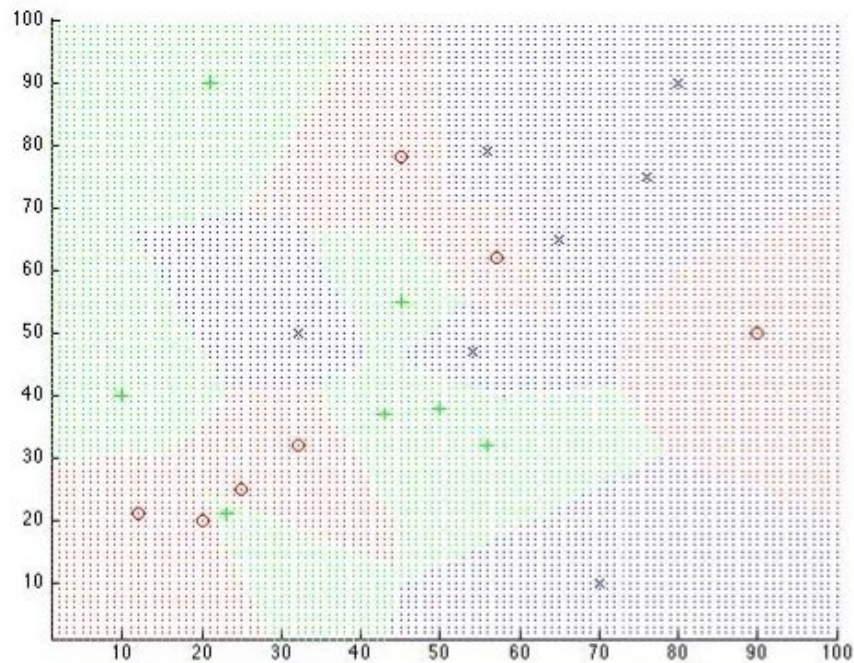


We will be using Euclidian distance to determine what are the k nearest neighbors:

$$d(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$



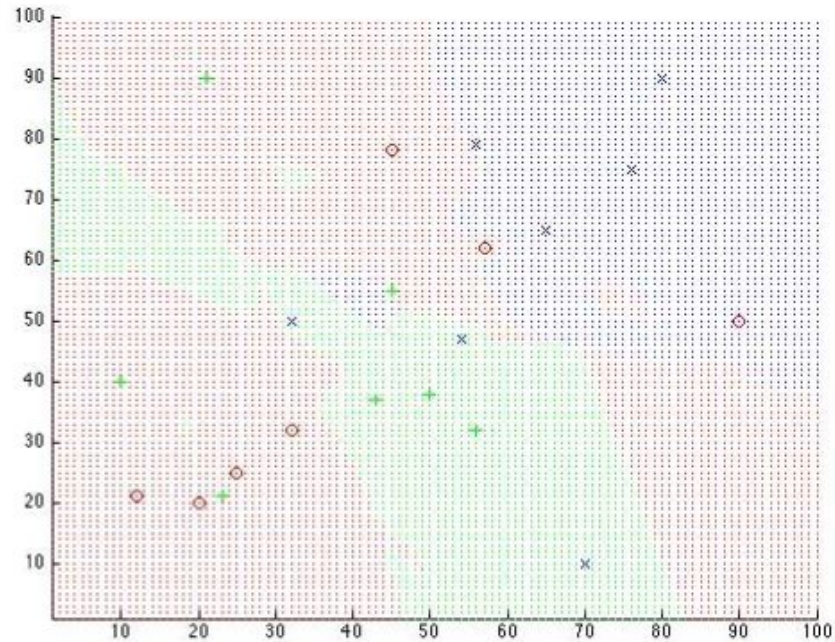
# The effect of increasing K



$K = 1$

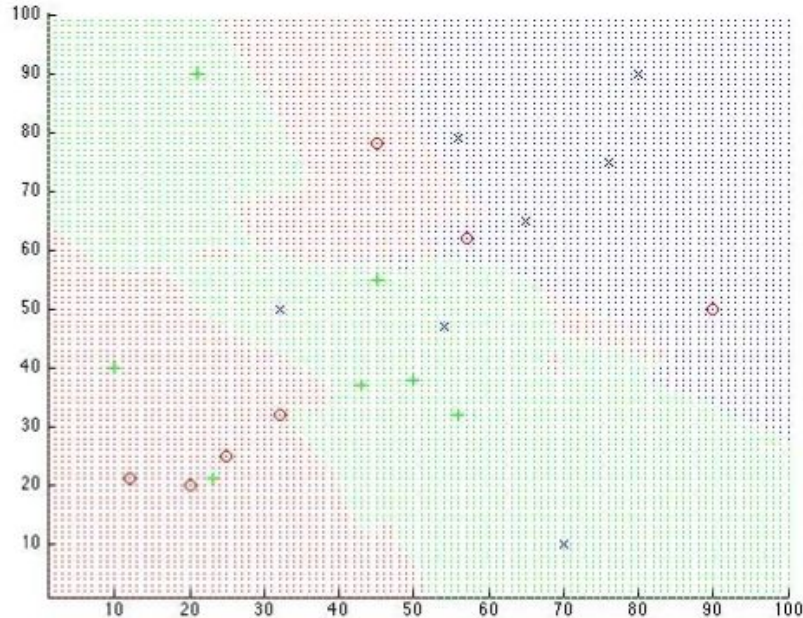


# The effect of increasing K



K = 3

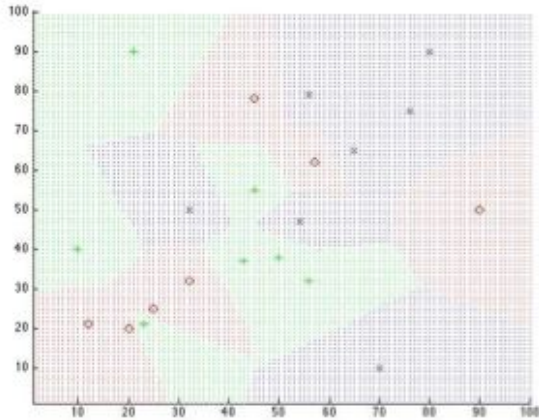
# The effect of increasing K



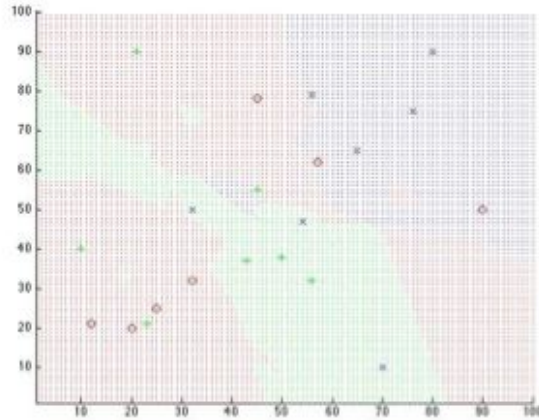
K = 5

# Comparison of different values of K

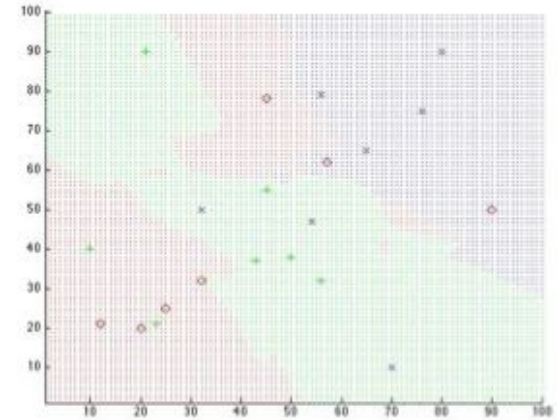
K = 1



K = 3



K = 5







# A probabilistic interpretation of KNN

- The decision rule of KNN can be viewed using a probabilistic interpretation
- What KNN is trying to do is approximate the Bayes decision rule on a subset of the data
- To do that we need to compute certain properties including the conditional probability of the data given the class ( $p(x|y)$ ), the prior probability of each class ( $p(y)$ ) and the marginal probability of the data ( $p(x)$ )
- These properties would be computed for some small region around our sample and the size of that region will be **dependent on the distribution of the test samples\***

- Let  $V$  be the volume of the  $m$  dimensional ball around  $z$  containing the  $k$  nearest neighbors for  $z$  (where  $m$  is the number of features).
- Then we can write

$$p(x)V = P = \frac{K}{N} \quad p(x) = \frac{K}{NV} \quad p(x | y = 1) = \frac{K_1}{N_1V} \quad p(y = 1) = \frac{N_1}{N}$$

- Using Bayes rule we get:

Choose the class with the highest probability

$$p(y = 1 | z) = \frac{p(z | y = 1)p(y = 1)}{p(z)} = \frac{K_1}{K}$$

$z$  – new data point to classify

$V$  - selected ball

$P$  – probability that a random point is in  $V$

$N$  - total number of samples

$K$  - number of nearest neighbors

$N_1$  - total number of samples from class 1

$K_1$  - number of samples from class 1 in  $K$





## Bayes decision rule

**Bayes Rule:**  $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

# Bayes decision rule

x – input feature set  
y - label

- If we know the conditional probability  $p(x | y)$  and class priors  $p(y)$  we can determine the appropriate class by using Bayes rule:

$$P(y = i | x) = \frac{P(x | y = i)P(y = i)}{P(x)} \stackrel{\text{def}}{=} q_i(x)$$

Minimizes our probability of making a mistake

- We can use  $q_i(x)$  to select the appropriate class.
- We chose class 0 if  $q_0(x) \geq q_1(x)$  and class 1 otherwise
- This is termed the ‘Bayes decision rule’ and leads to optimal classification.
- However, it is often very hard to compute ...

Note that  $p(x)$  does not affect our decision




## Bayes decision rule

$$P(y = i | x) = \frac{P(x | y = i)P(y = i)}{P(x)} \stackrel{def}{=} q_i(x)$$

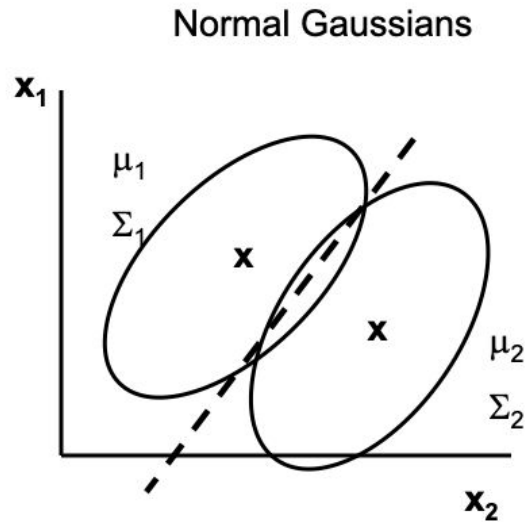
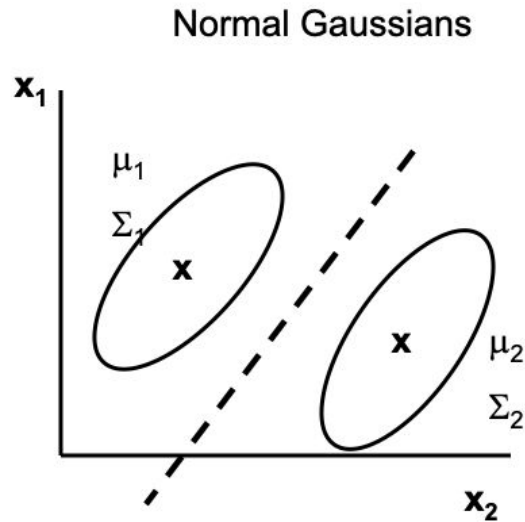
- We can also use the resulting probabilities to determine our **confidence** in the class assignment by looking at the likelihood ratio:

$$L(x) = \frac{q_0(x)}{q_1(x)}$$



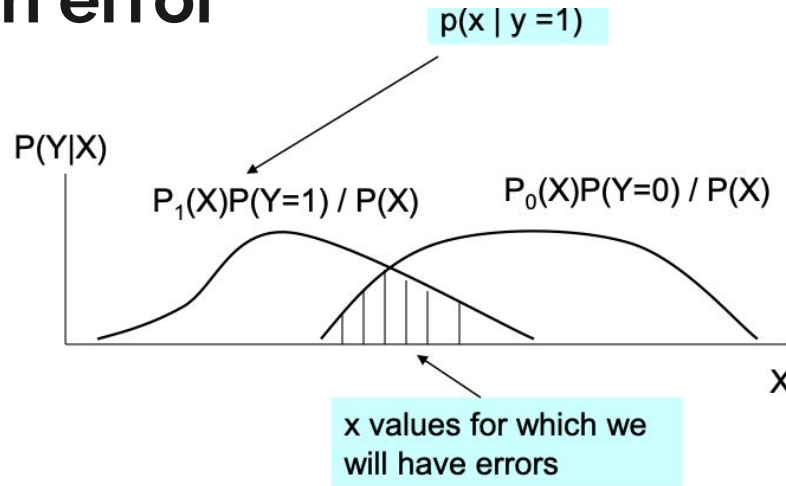
Also known as likelihood ratio, we will talk more about this later

# Binary case: separable v.s. non-separable



# Classification error

- For the Bayes decision rule we can calculate the probability of an error
- This is the probability that we assign a sample to the wrong class, also known as the **risk**



- The risk for sample  $x$  is:

$$R(x) = \min\{P_1(x)P(y=1), P_0(x)P(y=0)\} / P(x)$$

Risk can be used to determine a 'reject' region

# Bayes error

- The probability that we assign a sample to the wrong class, is known as the **risk**

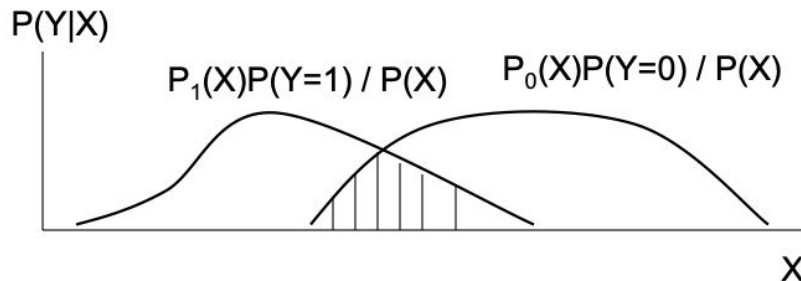
- The risk for sample  $x$  is:

$$R(x) = \min\{P_1(x)P(y=1), P_0(x)P(y=0)\} / P(x)$$

- We can also compute the expected risk (the risk for the entire range of values of  $x$ ):

$$\begin{aligned} E[r(x)] &= \int r(x)p(x)dx \\ &= \int \min\{p_1(x)p(y=1), p_0(x)p(y=0)\} dx \\ &= p(y=0) \int_{L_1} p_0(x)dx + p(y=1) \int_{L_0} p_1(x)dx \end{aligned}$$

$L_1$  is the region where we assign instances to class 1





# Takeaways

- Optimal decision using Bayes rule
- Type of classifiers
- Effect of values of  $K$  on KNN classifiers
- Probabilistic interpretation of KNN



# References

- Tom Mitchell: Machine Learning, Chapter 8
- Kevin Murphy: Machine Learning: A probabilistic perspective, Chapter 14
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Chapter 6, 13
- Ziv Bar-Joseph, Tom Mitchell, Pradeep Ravikumar and Aarti Singh: CMU 10-701