*k*NN
(*k*-Nearest Neighbors)

Boris Velichkov

Machine Learning

Supervised Learning

Unsupervised Learning

Classification

Regression

Clustering

Classification

Dataset

	Attribute 1	Attribute 2	…	…	Attribute N	Class
X1						„Yes“
.						„No“
. .						„Yes“
Xm						„Yes“

Xt = (a1, a2, … , aN)

Class(Xt) = ?

Regression

Dataset

	Attribute 1	Attribute 2	…	…	Attribute N	Class
X1						1.5
.						0.7
.						1.8
. Xm						1.2

Xt = (a1, a2, … , aN)

Class(Xt) = ?

Clustering

Dataset

	Attribute 1	Attribute 2	…	…	Attribute N
X1					
.					
.					
.					
Xm					

$\forall$ xi $\in$ D, xi = (a1, a2, … , aN)

D = (x1, x2, … , xm)

Clusters (D) = ?

Global Learning vs Local Learning

- **Global Learning**: Learning from all instances in the dataset.
  - Naïve Bayes Classifier
- **Local Learning**: Learning from some of the instances in the dataset.
  - *k*NN

***** *Local & Global Learning is different from Local & Global Search!*

Instance-Based Learning vs Model-Based Learning

- **Instance-Based Learning**: Use the entire dataset as the model.
  - *k*NN
- **Model-Based Learning**: Use the training data to create a model that has parameters learned from the training data.
  - Naïve Bayes Classifier

Lazy Learning vs Eager Learning

- Lazy vs. eager learning
  - Lazy learning (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
  - Eager learning (eg. Decision trees, SVM, NN): Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

Lazy Learner: Instance-Based Methods

- Instance-based learning:
  - Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified
- Typical approaches
  - k-nearest neighbor approach
    - Instances represented as points in a Euclidean space.
  - Locally weighted regression
    - Constructs local approximation
  - Case-Based Reasoning (CBR)

kNN

- All instances correspond to points in the n-D space
- The nearest neighbor are defined in terms of Euclidean distance, dist(X1, X2)
- Target function could be discrete- or real-value
- For discrete-valued, k-NN returns the most common value among the k training examples nearest to the test instance.

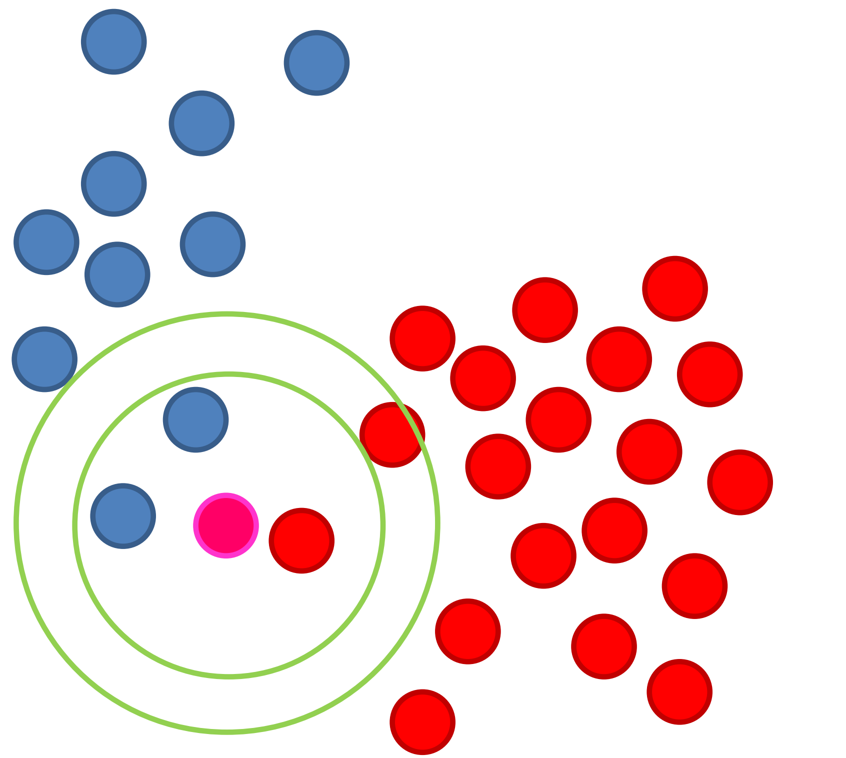
kNN

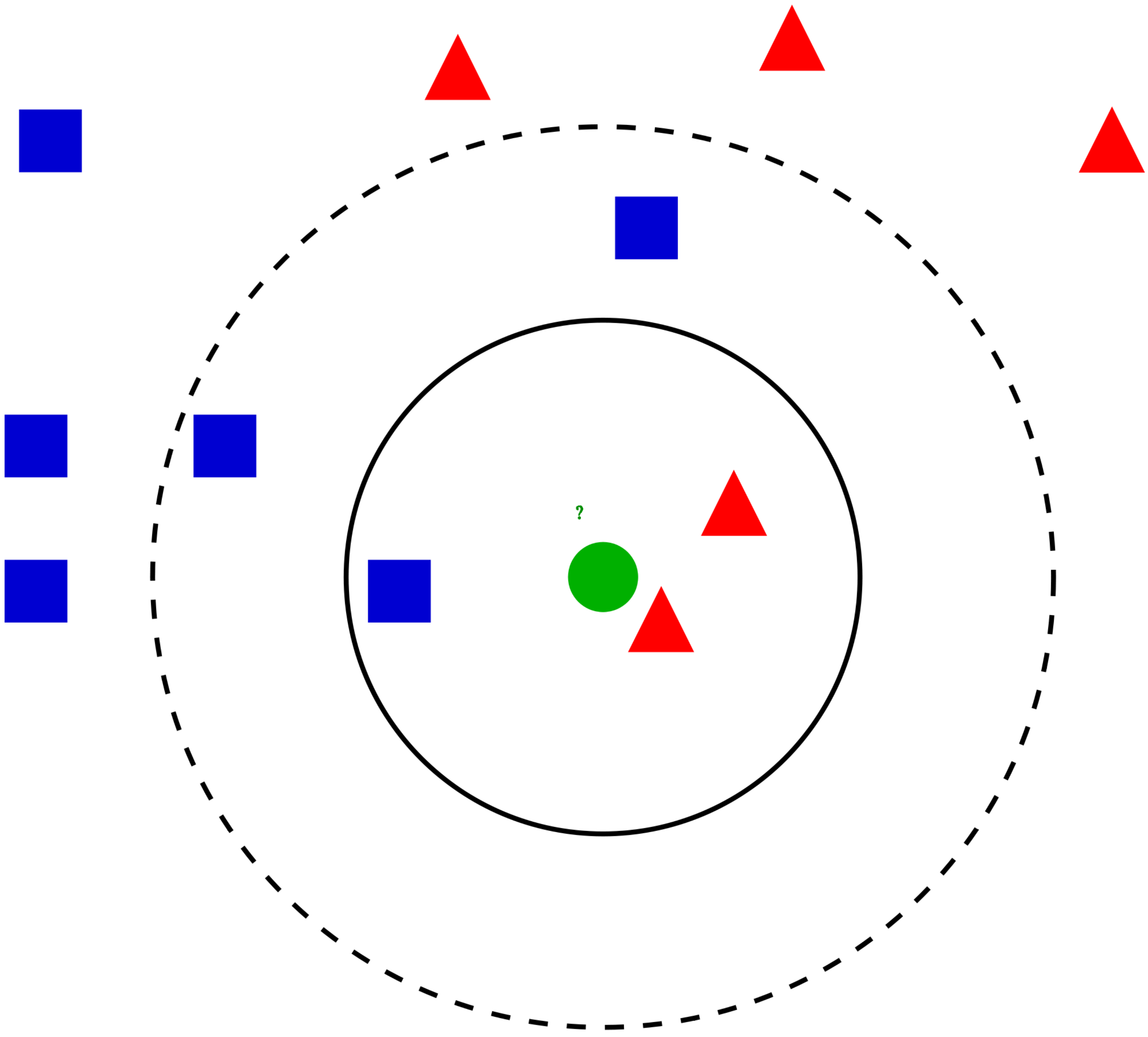
- k-NN for real-valued prediction for a given unknown tuple
  - Returns the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query xq
    - Give greater weight to closer neighbors
- Robust to noisy data by averaging k-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant

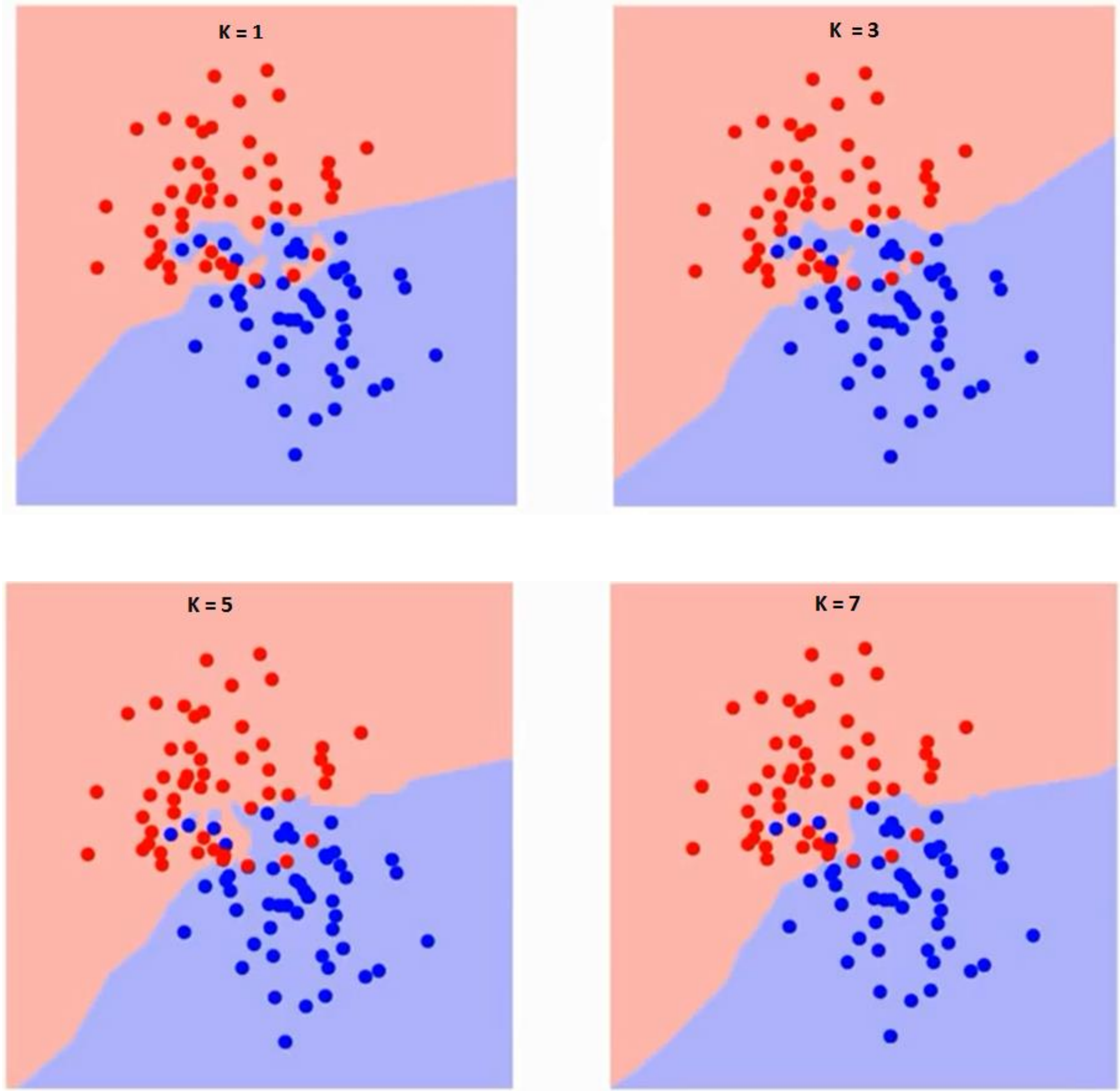
kNN

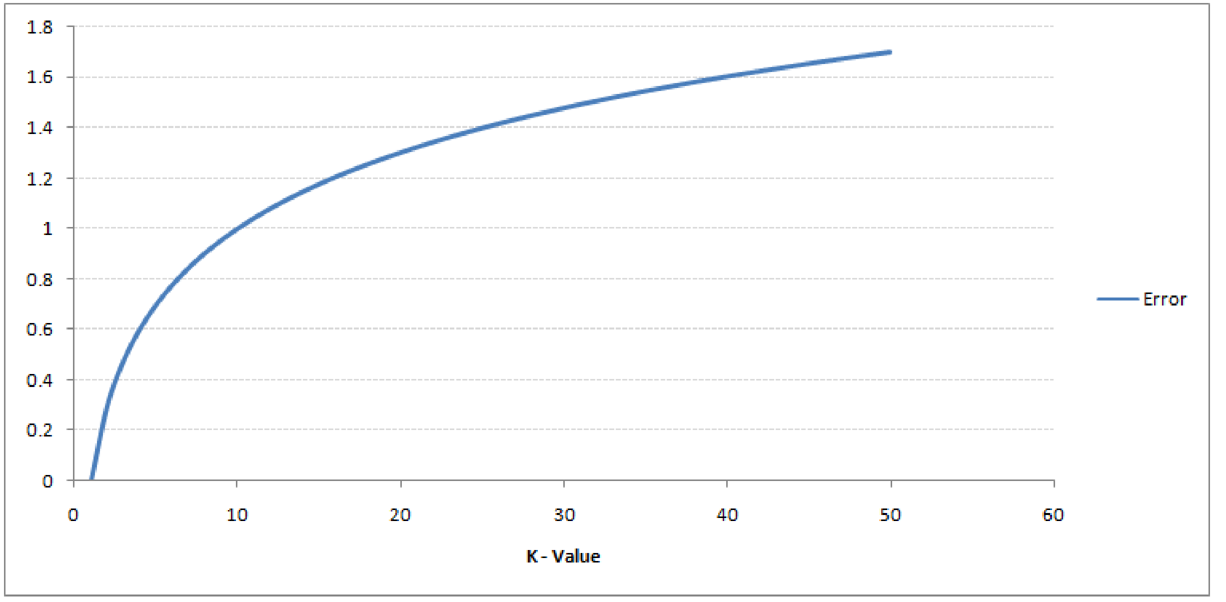
- Kernel estimation
  - k-nearest neighbor

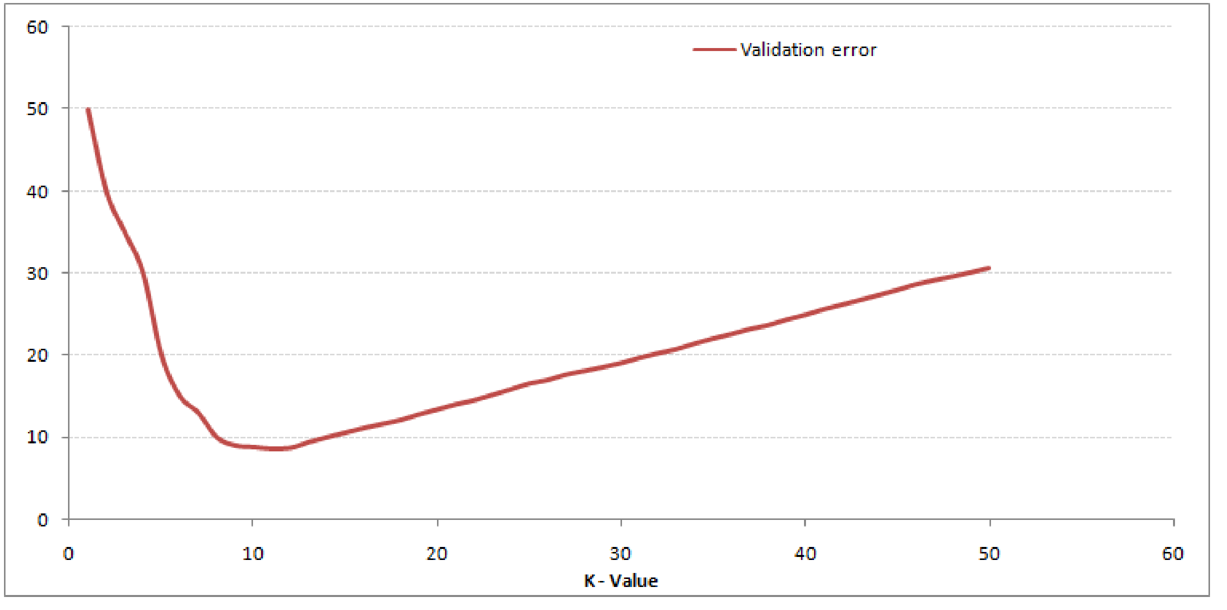
K = 3






K = 1
K = 3
K = 5
K = 7


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
K - Value
Error


60
50
40
30
20
10
0
0
10
20
30
40
50
60
K - Value
Validation error