Inspire…Educate…Transform.

Methods and Algorithms

Instance Based Learning - KNN

Dr. Suryaprakash Kompalli

Senior Mentor, INSOFE

K-NN

Collaborative filtering

Radial basis functions

INSTANCE BASED LEARNING

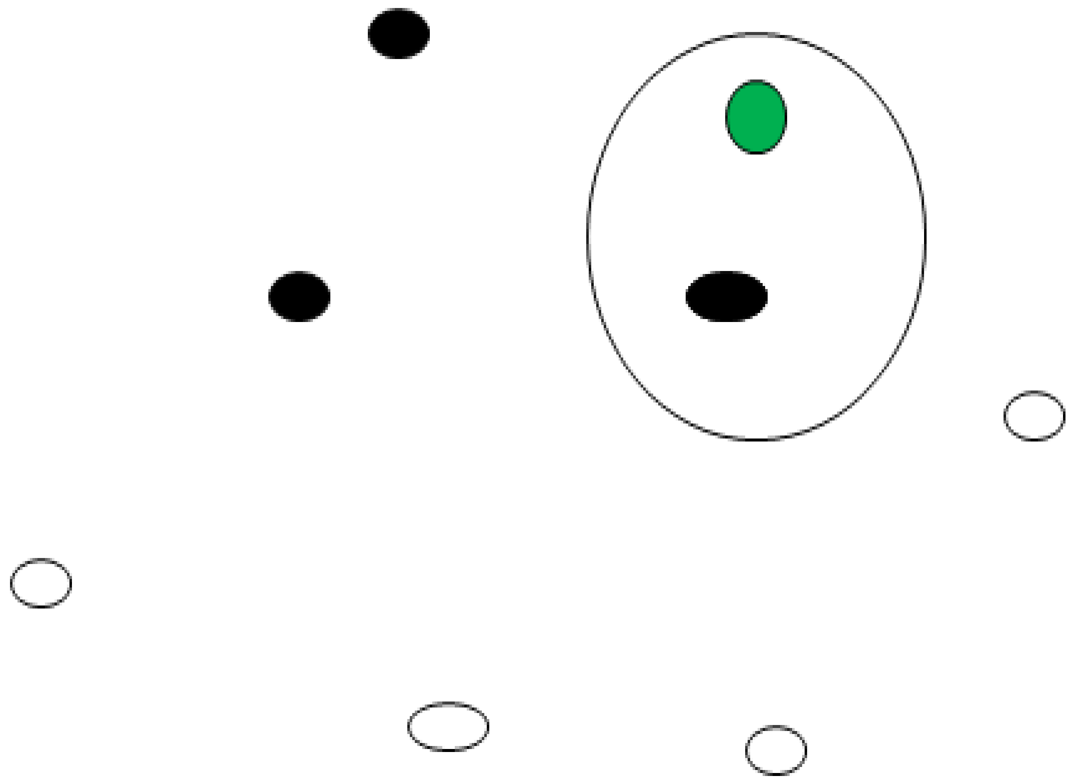
KNN-Algorithm

- Assign a class to a new data point based on its neighbors (mode)

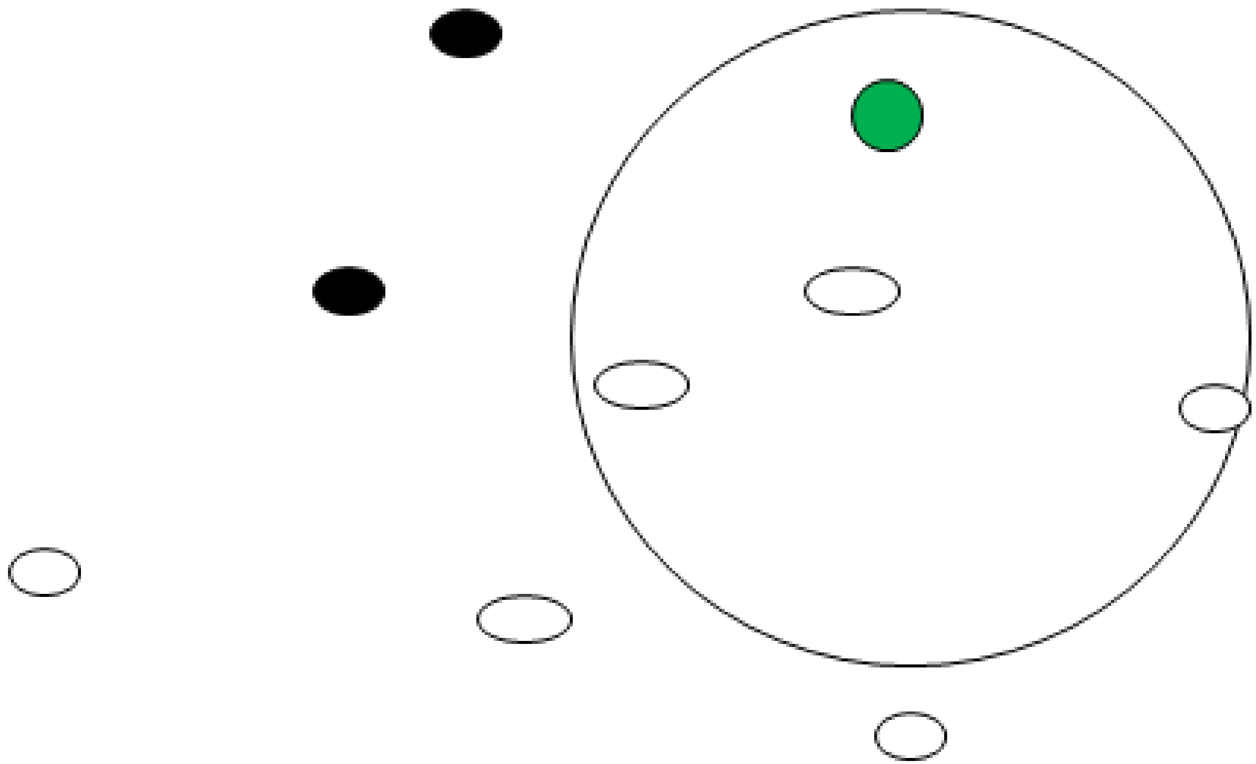
- Identify a numeric value of a new data point based on its neighbors (mean/median)

- Weighted mean/mode of entire data

K=1



K=3



Process is simple

- Pick a number of neighbors you want to use for classification or regression (K)
- Choose a method to measure distances (same consideration as clustering)
- Keep a data set with records

Process

- For every new point, identify the number of nearest neighbors you picked using the method you chose

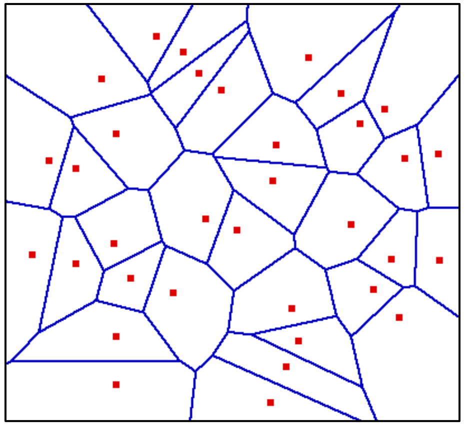
- Let them vote if it is a classification issue or take a mean/median for regression!

K-NN is

- Supervised
- Non parametric
- Lazy
- Local heuristic

Observations

- Decision surfaces created by KNN:
  - Voronoi Diagrams: Each point in a convex hull is closest to the sample inside the convex hull than to any other sample
  - Much more complex than decision trees!
    - http://www.raymondhill.net/voronoi/rhill-voronoi.html
    - http://www.pi6.fernuni-hagen.de/GeomLab/VoroGlide/



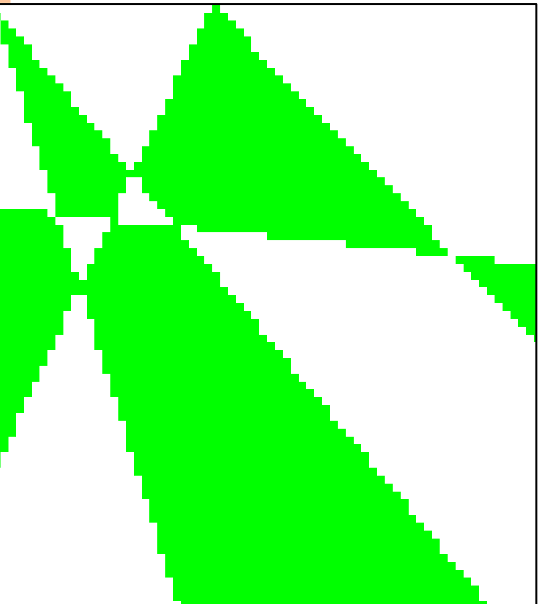
- Theoretical guarantee
  - Noisy training is an issue
  - Can overfit

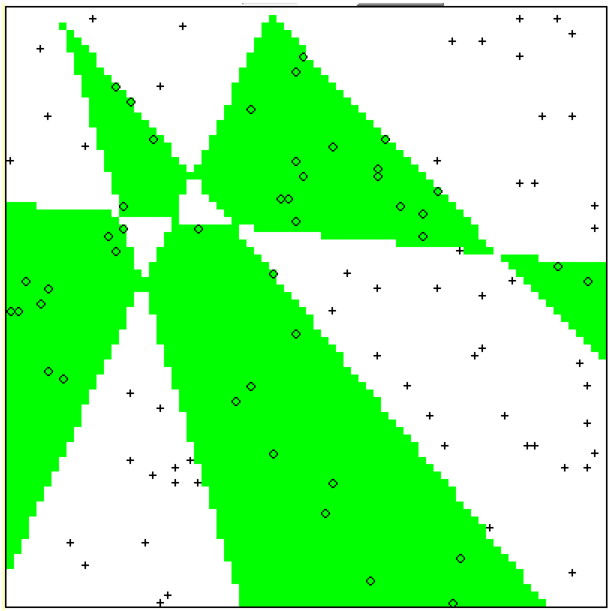
Let us play

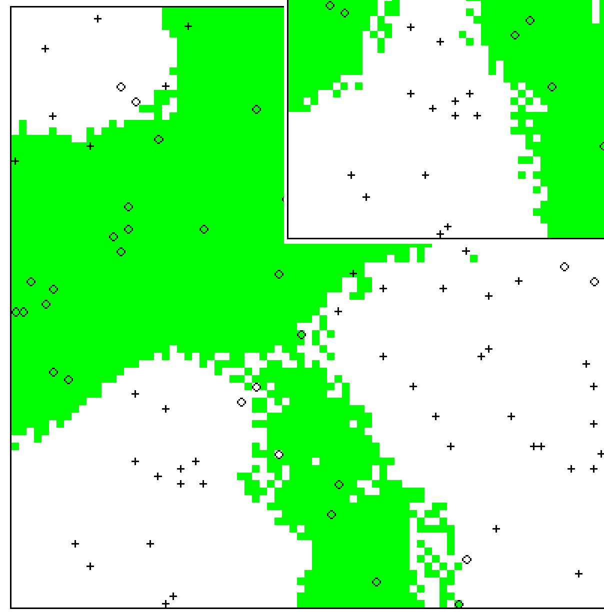
- http://sleepyheads.jp/apps/knn/knn.html
  - (Two class example)
- http://www.cs.cmu.edu/~zhuxj/courseproject/knndemo/KNN.html
  - Changing K allows you to model true distribution
- Sample applications:
  - http://siret.ms.mff.cuni.cz/sir/

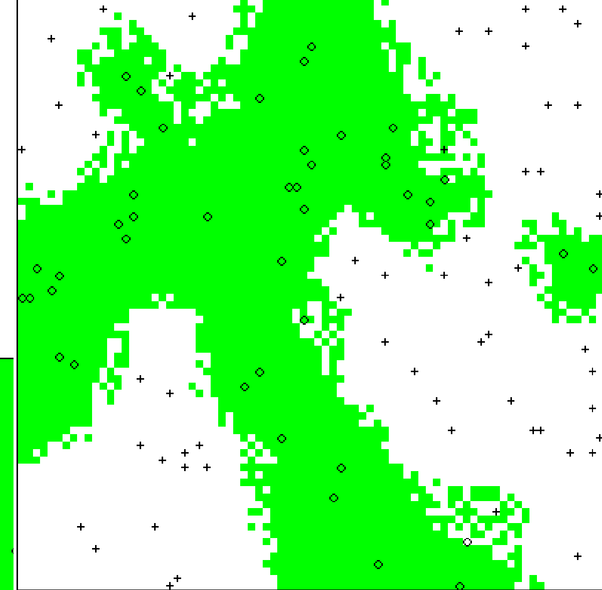
Behavior of KNN

http://www.cs.cmu.edu/~zhuxj/courseproject/knndemo/KNN.html









Decision surface, K=10

Decision s

- Experiment with K to see how decision surface is obtained

Observations

- As the complexity of space grows, accuracy comes down and you need more data

- Increasing K can reduce the overfit. There will be an optimum K

Issues with KNN and instance based techniques

- Curse of dimensionality
- Requires more memory and more time

Attributes

Records

Search process

ENGINEERING K-NN

Attributes

- Scaling the attributes is important
  - Attributes with larger range can dominate
  - Categorical variables and Ordinal variables need to be converted to numeric
    - Think Value Distance Metric

Attributes: Distance metric

Value difference measure (VDM):

$$\sum_{h=1}^{All\ classes} |P(h|val_i) - P(h|val_j)|$$

CSE 7301c

Attributes: Value Distance Measure

ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

$$VDM_{family1,family2}$$

$$|P(0|f_1) - P(0|f_2)| + |P(1|f_1) - P(1|f_2)|$$

$$|0.5 - 0| + |0.5 - 1|$$

$$= 1$$

$$VDM_{family1,family3} = |P(0|f_1) - P(0|f_3)| + |P(1|f_1) - P(1|f_3)| = |0.5 - 0.5| + |0.5 - 0.5| = 0$$

Curse of dimensionality

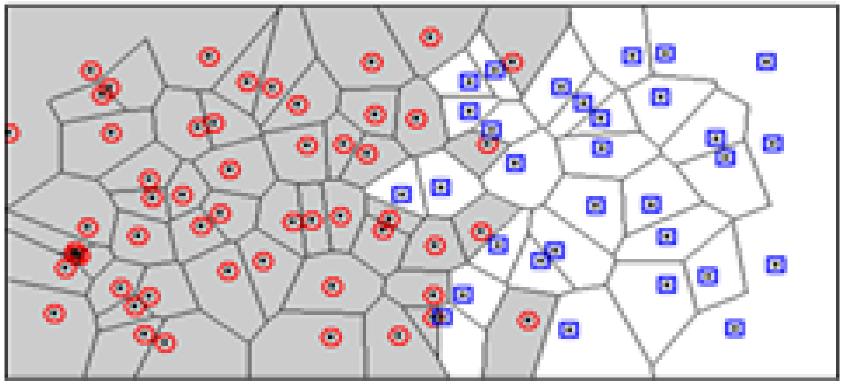
- K-NN is heavily impacted as all points are at the surface and hence similar
- Reduce the dimensions
  - Correlation
  - Info gain (Can lose some that are important; it assumes independent attributes)
    - Wrapper methods
      - Forward selection, Backward elimination
  - Weighting attributes
    - Think PCA

R-KNN

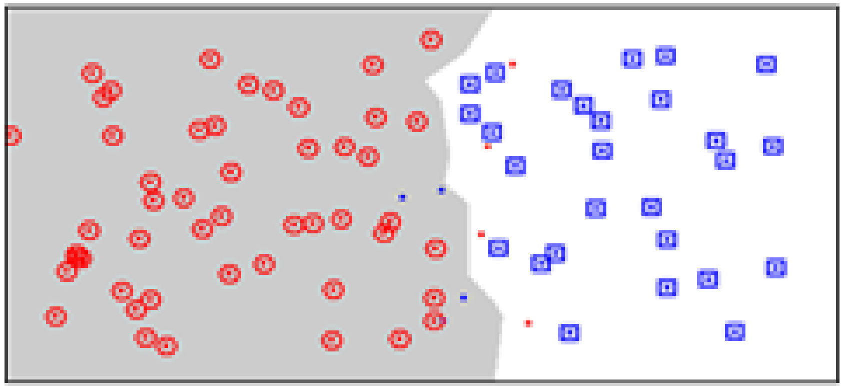
- The same concept of randomforest but for k-nn
  - http://www.biomedcentral.com/1471-2105/12/450
- Can be used for feature selection

Wilson editing

Overlapping classes



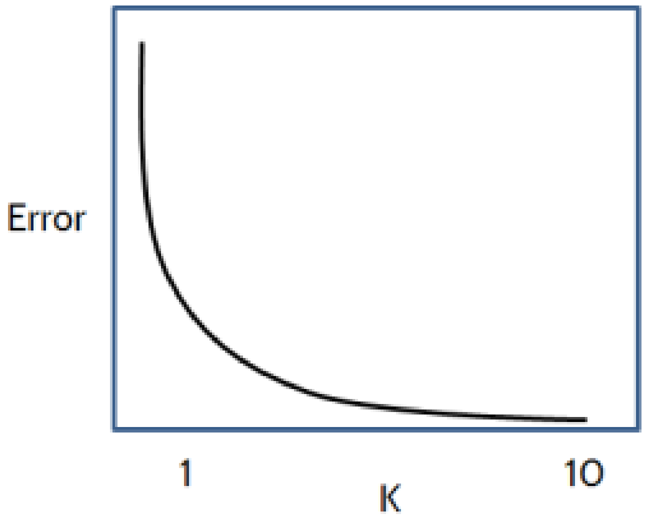
Original data

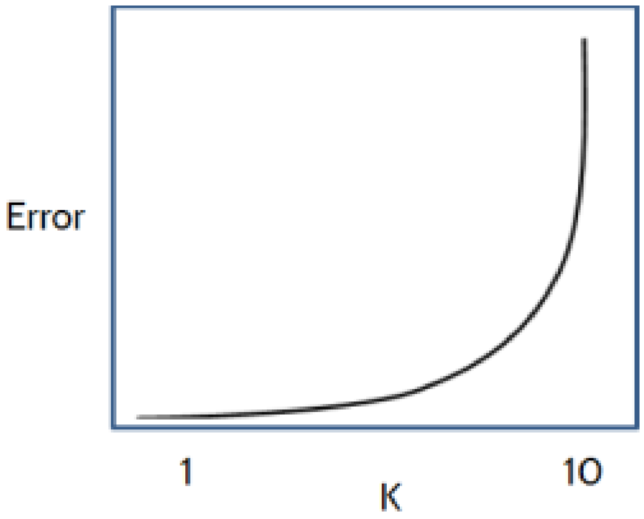


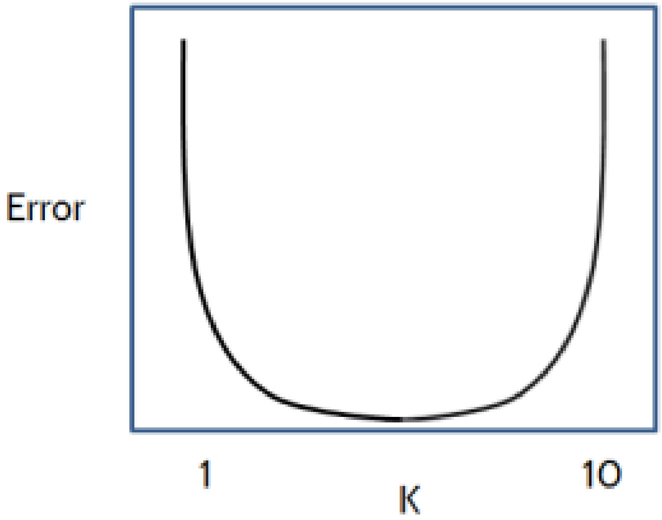
Wilson editing with k=7

One method for outlier removal
Remove points that do not agree with the majority of their k nearest neighbours

Use a K that gives least error on test data







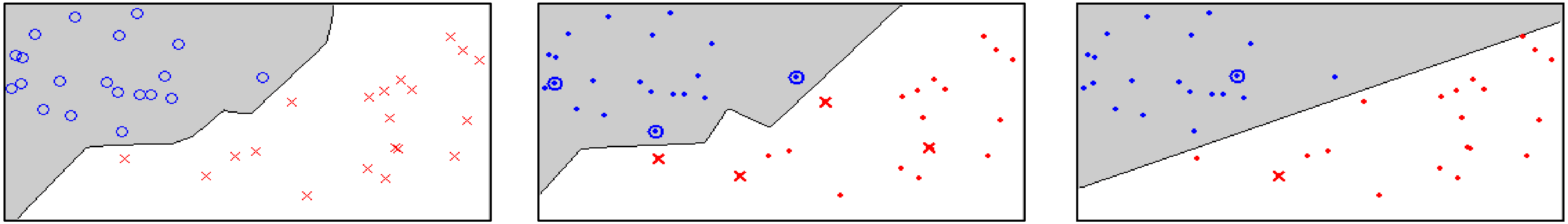
Records: Handling missing values

- K-NN is impacted heavily by missing values

- Imputation is one option but might be self defeating

Speeding up search

Delaunay triangulation

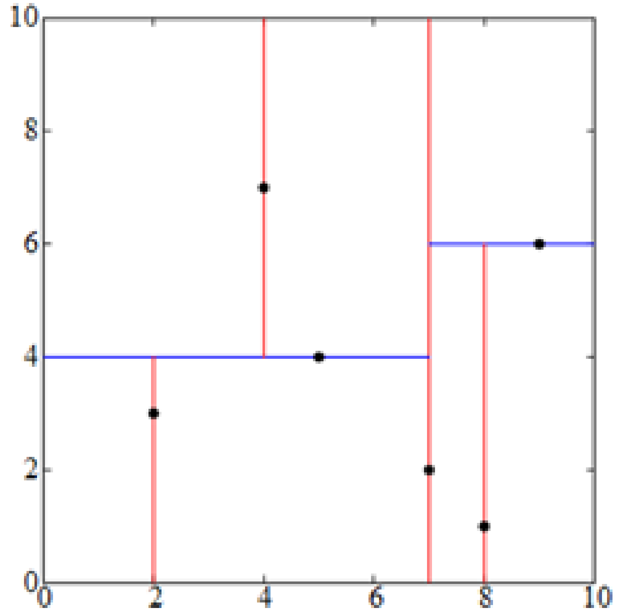


Original data **Condensed data** **Minimum Consistent Set**

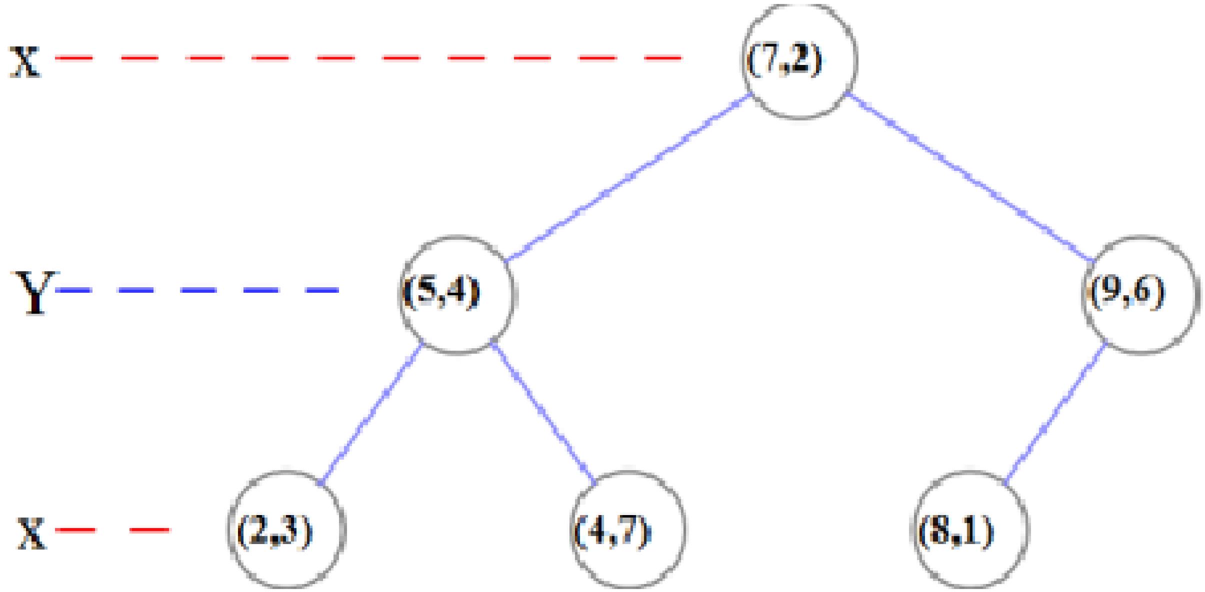
http://www.cse.unsw.edu.au/~lambert/java/3d/delaunay.html

Cran library: Class

Kd Tree (CRAN: FNN)

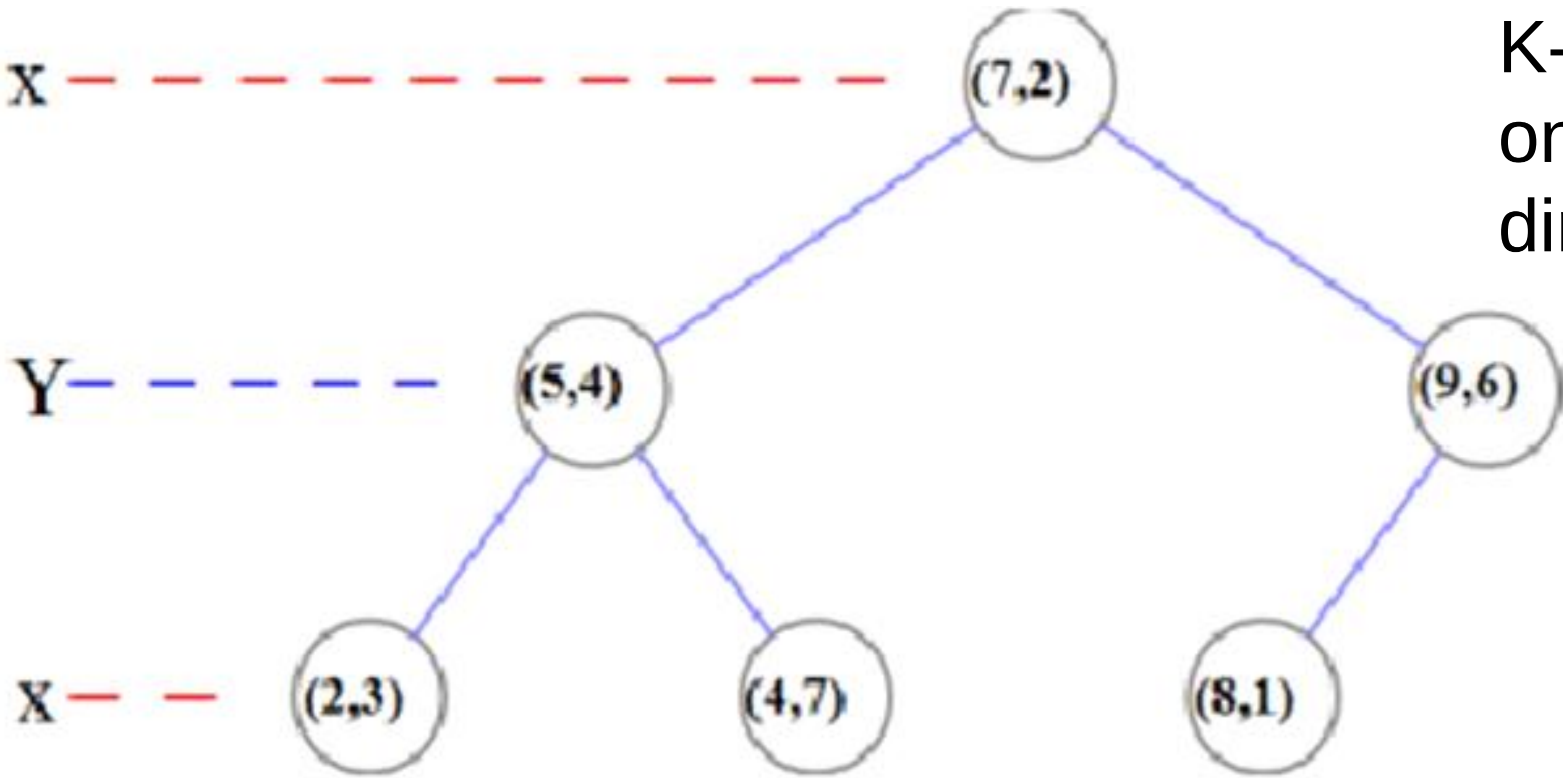


K-d tree decomposition for the point set (2,3), (5,4), (9,6), (4,7), (8,1), (7,2)



Resulting K-d tree

Searching NN for (3,3)



K-D tree works only for low dimensions

Speeding up 3

- Come up with rough approximations to eliminate most points (distance between centroids) and then apply elaborate measurements on closest points

Summary

- Scaling the data
- Address dimensionality:
    - Correlation
    - Coarse approximations
    - R-KNN
- Remove outliers
    - Condensation of data using Delaunay Triangulation
- Use intelligent data structures: KD Tree

COLLABORATIVE FILTERING

Collaborative filtering

- How do I recommend?
  - Association rules
  - Similarity based (collaborative filtering)
  - Model based

Collaborative filtering: primitive

Primitive version:

$$\hat{R}_{ik} = \alpha \sum_{X_j \in \mathbf{N}_i} W_{ij} R_{jk}$$

N_i is the space of users
R_{jk} is the rating of user j on movie k

$$\alpha = \left(\sum |W_{ij}|\right)^{-1}$$

Similarity (Pearson coefficient):

$$W_{ij} = \frac{\sum_k (R_{ik} - \overline{R}_i)(R_{jk} - \overline{R}_j)}{\sqrt{\sum_k (R_{ik} - \overline{R}_i)^2 (R_{jk} - \overline{R}_j)^2}}$$

Looks similar to correlation?

Collaborative filtering: More refined

$$\hat{R}_{ik} = \overline{R}_i + \alpha \sum_{X_j \in \mathbf{N}_i} W_{ij}(R_{jk} - \overline{R}_j)$$

Average rating of user “j” on all movies. Takes care of the difference in rating between user. Some users rate things consistently higher than others.

Collaborative filtering

	Matrix	Star Wars	Dark knight	Rocky	Sita Aur Gita	Star Trek	Cliffhanger	A.I.	MI	X-Men
Jim	1	3	1	5	2	1			1	
Sean	2		3	2		4		5		3
John		3		4		5			3	4
Sidd	4				3		4		2	
Penny	5		2		2		5		1	
Pete		5			?		4			4

Collaborative filtering

	Matrix	Star Wars	Dark knight	Rocky	Sita Aur Gita	Star Trek	Cliffhanger	A.I.	MI	X-Men
Jim	-0.65	0.65	-0.65	1.96	0	-0.65			-0.7	
Sean	-1		-0.14	-1		0.71		1.57		-0.14
John		-1		0.24		1.434			-1	0.24
Sidd	0.783				-0.26		0.78		-1.3	
Penny	1.069		-0.53		-0.53		1.07		-1.1	
Pete		1.15			?		-0.6			-0.58

Standardized rating: $\hat{x} = (x - \mu)/\sigma$

Estimate Unknown Rating

	Jim	Sean	John	Sidd	Penny	Pete
Jim	1.00	-0.31	-0.10	0.08	0.07	0.22
Sean	-0.31	1.00	0.17	-0.20	-0.22	0.03
John	-0.10	0.17	1.00	0.36	0.26	-0.44
Sidd	0.08	-0.20	0.36	1.00	0.93	-0.18
Penny	0.07	-0.22	0.26	0.93	1.00	-0.22
Pete	0.22	0.03	-0.44	-0.18	-0.22	1.00

	Matrix	Star Wars	Dark knight	Rocky	Sita Aur Gita	Star Trek	Cliffhanger	A.I.	MI	X-Men
Jim	-0.65	0.65	-0.65	1.96	0	-0.65			-0.7	
Sean	-1		-0.14	-1		0.71		1.57		-0.14
John		-1		0.24		1.434			-1	0.24
Sidd	0.783				-0.26		0.78		-1.3	
Penny	1.069		-0.53		-0.53		1.07		-1.1	
Pete		1.15			?		-0.6			-0.58

- q(Pete, SitaAurGita)
- Jim, Sidd and Penny rated SitaAurGita
- Jim and Sidd are the 2 closest neighbors to Pete based on our distance metric.
- The average of the ratings by Jim and Sidd to movie “SitaAurGita” is “-0.26”.

- We need to bring it back to Pete’s prediction level by multiplying by Standard Deviation of Pete’s ratings and adding Pete mean rating to this product. Approximately this is 1.603. A better thing would have been to take distance weighted mean. We will do this in the larger example.

Project

- Study the papers
  - http://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf
  - http://blog.yhathq.com/posts/recommender-system-in-r.html
  - http://www2.research.att.com/~volinsky/papers/ieeecomputer.pdf

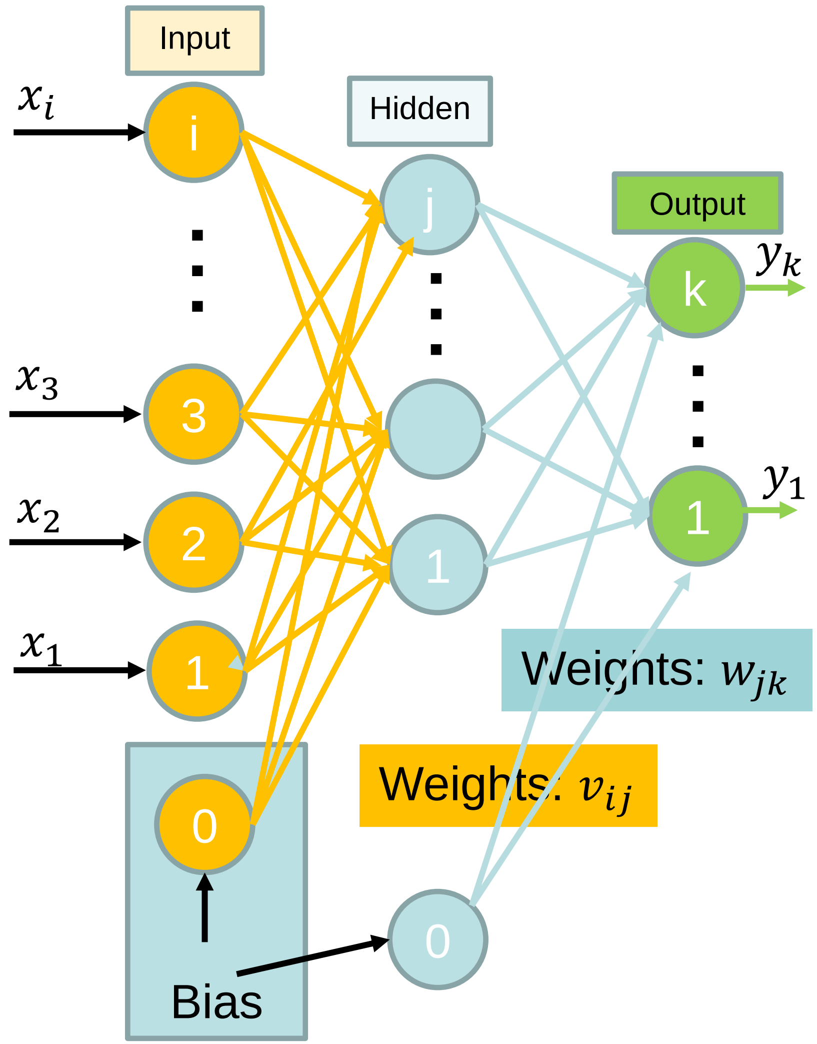
Evaluate at least three models

- Use recommender lab
- Songs data

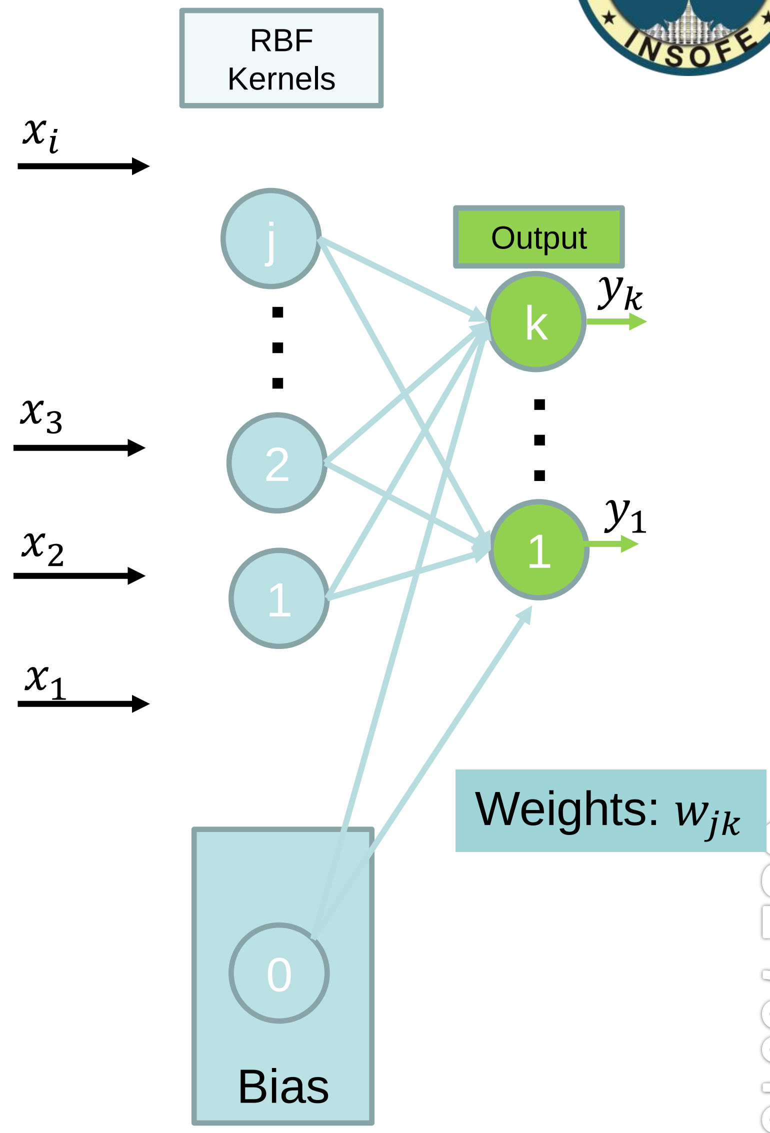
- Can you ensemble and do better than any individual model

RADIAL BASIS FUNCTIONS

Radial basis functions (CRAN: RSNNS)



Neural network



RBF network

Radial basis functions (CRAN: RSNNS)

Neural network output:

$$h_j = f\left(\sum_{i=0}^{I} x_i v_{ij}\right)$$

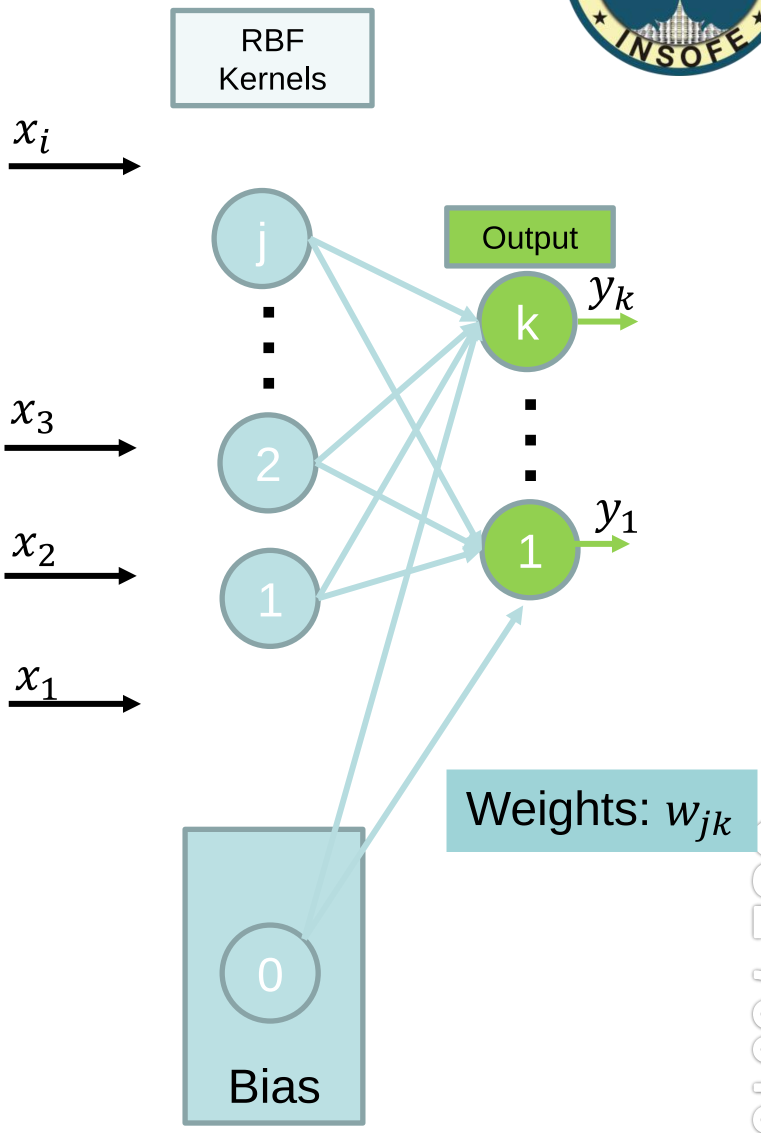
$$y_k = f\left(\sum_{j=0}^{J} h_j w_{jk}\right)$$

RBF network output:

$$h_j = g(x, \mu_j)$$

$$y_k = f\left(\sum_{j=0}^{J} h_j w_{jk}\right)$$

$$g:\ kernel\ function$$



RBF network

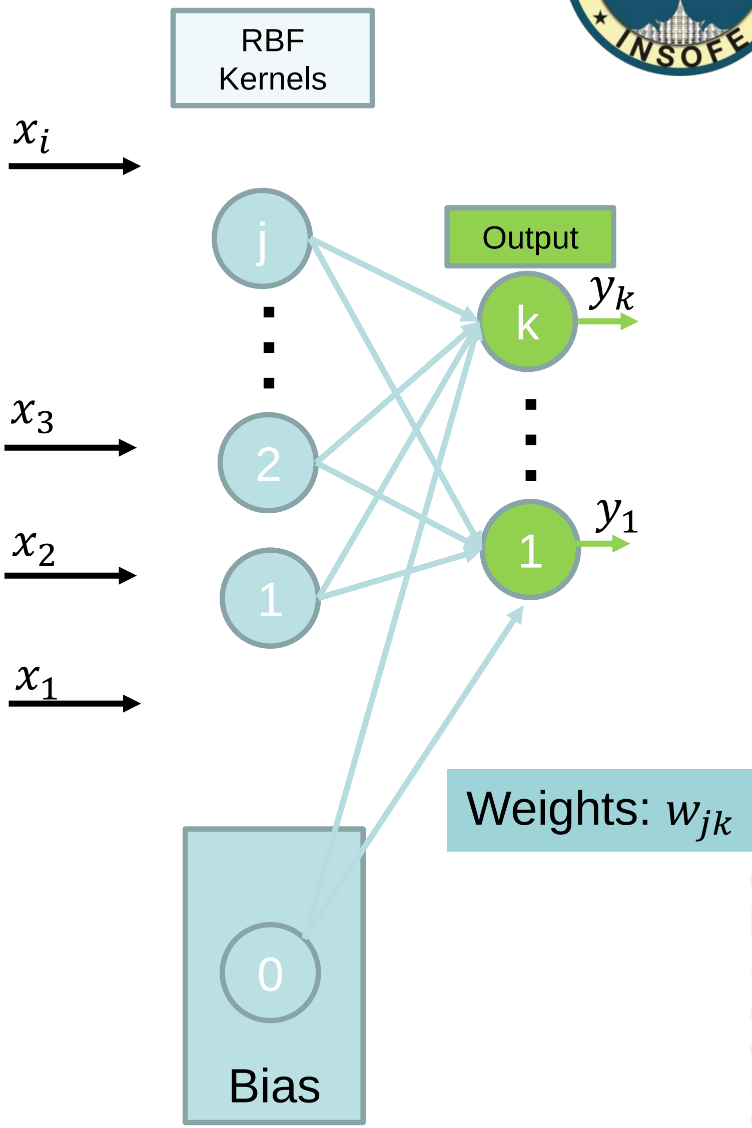
Radial basis functions (CRAN: RSNNS)

RBF network output:
$$h_j = g(x, \mu_j)$$
$$y_k = f\left(\sum_{j=0}^{J} h_j w_{jk}\right)$$

g: *kernel function*

- Choice of Kernel function:
  - Kmeans cluster centers
  - Gaussian: $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Training output weights w_{jk}
  - Back-propagation

Class question: Why called radial basis functions?



RBF network

RBF References

- https://chrisjmccormick.wordpress.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/
- http://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf
- http://research.cs.tamu.edu/prism/lectures/pr/pr_l19.pdf

International School of Engineering

2-56/2/19, Khanamet, Madhapur, Hyderabad - 500 081

For Individuals: +91-9177585755 or 040-65743991

For Corporates: +91-9618483483

Web: http://www.insofe.edu.in

Facebook: https://www.facebook.com/insofe

Twitter: https://twitter.com/Insofeedu

YouTube: http://www.youtube.com/InsofeVideos

SlideShare: http://www.slideshare.net/INSOFE

LinkedIn: http://www.linkedin.com/company/international-school-of-engineering

Radial Basis Function Networks

ire...Educate...Transform.

- Global approximation to target function, in terms of linear combination of local approximations
- Used, e.g., for image classification
- A different kind of neural network
- Closely related to distance-weighted regression, but "eager" instead of "lazy"

)81

Facebook: https://www.facebook.com/insofe

Twitter: https://twitter.com/Insofeedu

YouTube: http://www.youtube.com/InsofeVideos

SlideShare: http://www.slideshare.net/INSOFE

LinkedIn: http://www.linkedin.com/company/international-school-of-engineering

Inspire...Educate...Transform.

- http://www.dfki.uni-kl.de/~aabecker/Mosbach/Bergmann-CBR-Survey.pdf

Lab today

- On German credit
  - manually design rknn! for classification
  - Predict a numeric variable with
    - local regression and global regression
    - radial basis functions and global regression

PLR

- Find the nearest neighbors of a new data point
- Build a linear or quadratic regression locally

- Extremely useful when global models are failing