INSOFE
Inspire...Educate...Transform.






Inspire…Educate…Transform.

Predictive Analytics

Association Mining

Lt. Surya Kompalli
Senior Mentor, INSOFE

Who purchased the basket?

It is not over yet

- Most likely he/she is a vegetarian!
- He/she has been exposed to some foreign culture (how many Indians eat pickles!)

Market basket analysis

- Provides insight into which products tend to be purchased together and which are most amenable to promotion.

MB can give

- Actionable rules
- Trivial rules
  - People who buy shoes also buy socks
- Inexplicable
  - People who buy shirts also buy milk

It is not just retail and baskets

- Unusual combinations of insurance claims can be a sign of fraud and can spark further investigation.

- Medical patient histories can give indications of likely complications based on certain combinations of treatments.

- Text categorization

Transform the data

ORDER ID	LINE ITEM ID	B
ORDER ID	LINE ITEM ID	C

	PRODUCT A	PRODUCT B	PRODUCT C	PRODUCT D
ORDER ID	0	1	1	0

Correlation table (co - occurrence table)

	Product A	Product B	Product C	Product D
Product A				
Product B				
Product C				
Product D				

Let us do a problem

Product table

ID	Product
1	Orange juice
2	Soda
3	Milk
4	Window cleaner
5	Detergent

Line item table

ID	Order ID	Product ID	Quantity
1	1	1	2
2	1	2	1
3	2	3	3
4	2	1	2
5	2	4	1
6	3	1	2
7	3	5	3
8	4	1	1
9	4	5	1
10	4	2	2
11	5	2	2
12	5	4	3

Order ID	Products
1	Orange juice, Soda
2	Milk, orange juice, window cleaner
3	Orange juice, detergent
4	Orange juice, detergent, soda
5	Window cleaner, soda

Co-occurrence

Product	Orange juice	Window cleaner	Milk	Soda	Detergent
Orange juice					
Window cleaner					
Milk					
Soda					
Detergent					

Co-occurrence

Product	OJ	Window Cleaner	Milk	Soda	Detergent
OJ	4	1	1	2	2
Window cleaner	1	2	1	1	0
Milk	1	1	1	0	0
Soda	2	1	0	3	1
Detergent	2	0	0	1	2

Insights

- Orange juice and soda are more likely to be purchased together than any other two items.
- Detergent is never purchased with window cleaner or milk.
- Milk is never purchased with soda or detergent.

Question

- How do we generate these rules automatically on large data

APRIORI ALGORITHM

Closure property

- A set is said to be closed under an operation, if the operation produces another member of the set in all situations.

Set of natural numbers

- is closed under
  - Addition
  - Multiplication.
- Not closed under
  - Subtraction and
  - Division

Downward Closure

- A set is said to be downward closed under a property if all its subsets also are closed.

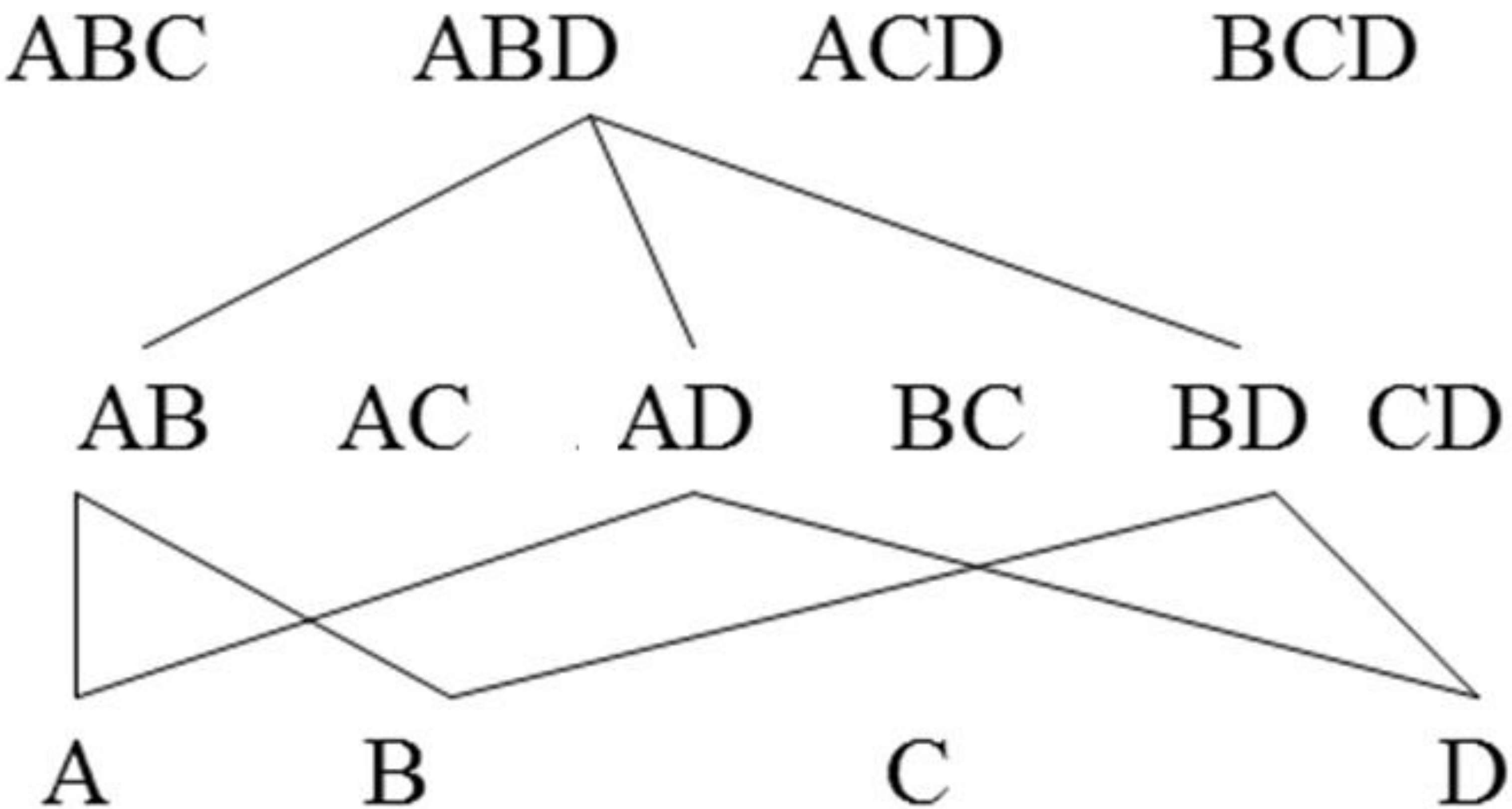
- Frequent itemsets have the downward closure property.

- Suppose {A,B} has a certain frequency (f). Since each occurrence of A,B includes both A and B, then both A and B must also have frequency >= f.

- Similar argument for larger itemsets

- So, if a k-itemset meets a cut-off frequency, all its subsets (k-1, k-2 itemsets) also meet this cut-off frequency

Downward closure



The Apriori Algorithm — Example

Database D
Minsup = 0.5

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2/4
{2}	3/4
{3}	3/4
{4}	1/4
{5}	3/4

L_1

{1}	2/4
{2}	3/4
{3}	3/4
{5}	3/4

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

{1 2}	1/4
{1 3}	2/4
{1 5}	1/4
{2 3}	2/4
{2 5}	3/4
{3 5}	2/4

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2/4

The Apriori Algorithm — Example

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

- How to prioritize associations:

itemset	sup
{1 3}	2/4
{2 3}	2/4
{2 5}	3/4
{3 5}	2/4

Antecedent	Consequent	Support	Confidence	Lift
{2}	{3,5}	3/4	2/3	conf({2}->{3,5})/prob(3,5)=0.67/0.5
{3}	{2,5}	3/4	2/3	
{5}	{2,3}	3/4	2/3	
{2,3}	{5}	2/4	2/2	conf({2,3}->{5})/prob(5)=1/0.75
{3,5}	{2}	2/4	2/2	
{2,5}	{3}	2/4	2/3	

Support: probability of Antecedent occurring
Confidence: Number of times antecedent and consequent were occurring together/number of times Antecedent was present
Lift: Confidence / Prob of the Consequent

Details: the algorithm

Algorithm Apriori(*T*)

$C_1 \leftarrow$ init-pass(T);

$F_1 \leftarrow \{f \mid f \in C_1, f.\text{count}/n \geq minsup\}$; // n: no. of transactions in T

for ($k = 2$; $F_{k-1} \neq \emptyset$; k++) **do**

$C_k \leftarrow$ candidate-gen(F_{k-1});

for each transaction $t \in T$ **do**

for each candidate $c \in C_k$ **do**

if c is contained in t **then**

c.count++;

end

end

$F_k \leftarrow \{c \in C_k \mid c.count/n \geq minsup\}$

end

return $F \leftarrow \bigcup_k F_k$;

Candidate-gen function: Join, Prune

Function candidate-gen($F_{k\text{-}1}$)

$C_k \leftarrow \emptyset$;

for all $f_1, f_2 \in F_{k\text{-}1}$

with $f_1 = \{i_1, \ldots, i_{k\text{-}2}, i_{k\text{-}1}\}$

and $f_2 = \{i_1, \ldots, i_{k\text{-}2}, i'_{k\text{-}1}\}$

and $i_{k\text{-}1} < i'_{k\text{-}1}$ **do**

$c \leftarrow \{i_1, \ldots, i_{k\text{-}1}, i'_{k\text{-}1}\}$; // join f_1 and f_2

$C_k \leftarrow C_k \cup \{c\}$;

for each (k-1)-subset s of c **do**

if ($s \notin F_{k\text{-}1}$) **then**

delete c from C_k; // prune

end

end

return C_k;

An example

- $F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$

- After join
  - $C_4 = \{\{1, 2, 3, 4\}, \{1, 3, 4, 5\}\}$
- After pruning:
  - $C_4 = \{\{1, 2, 3, 4\}\}$

    because $\{1, 4, 5\}$ is not in F_3 ($\{1, 3, 4, 5\}$ is removed)

Limitations

- *Apriori* algorithm can be very slow and the bottleneck is candidate generation.
  - For example, if the transaction DB has 10K frequent 1-itemsets, they will generate 10,000K candidate 2-itemsets even after employing the downward closure.
  - To compute those with sup more than minsup, the database need to be scanned at every level. It needs (n +1) scans, where n is the length of the longest pattern.

Radial Basis Functions

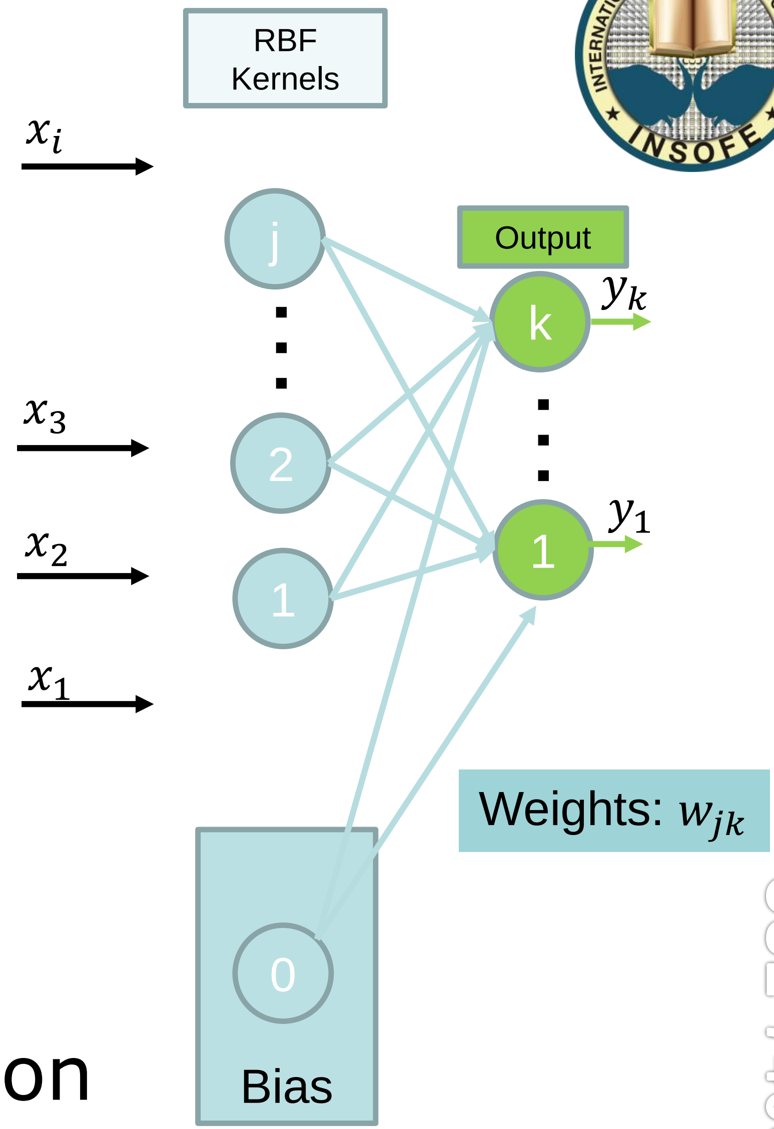
RBF network output:

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

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

$$g\text{: } kernel\ function$$

- A form of "Stacking"
  - "g" can be an RBF
    - K-means cluster center
    - Gaussian
    - Any other weak classifier
  - Train w_{jk} using back-propagation
  - RBF kernel SVM?
    - Good for images/real values, bad for text



RBF network

Looking back at Kernels

- Gaussian (radial-basis function network):

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

Use this when attributes need to be smoothed out

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0 \mathbf{x_i}^{\mathbf{T}} \mathbf{x_j} + \beta_1)$

Use this when an attribute has two distinct levels

https://charlesmartin14.wordpress.com/2012/02/06/kernels_part_1/

ML Matrix

	Sparse data			Dense data		
	Low noise	Moderate noise or outlier	Noise and outlier	Low noise	Moderate noise or outlier	Noise and outlier
SVM	Good	Soft margin SVM		Good		Not good
Neural net	Moderate			Good		
K-Nearest Neighbor						
Logistic regression						
Decision Tree				Not good if all attributes are continuous		

Fill in good, moderate, not good in above table. Feel free to add comments.

CSE 7405c

Summary

Understanding Error

- Theoretically, if Bayes error is e, KNN error will be at most 2e
- Given a dataset, try KNN and get error $\hat{e}$
- Try another classifier, if error is less than $\frac{\hat{e}}{2}$, you are in luck
- Many classifiers have bias
  - Logistic, SVM: Linear distribution
  - Gaussian: Normal distribution

Regression

- Linear and Logistic are linear models
  - Biased towards linear data
- If data not linearly separable:
  - Use Kernel techniques
  - Popular kernels: Polynomial, Radial basis, Gaussian
  - Kernel must be positive semi-definite function (Continuously differentiable)
  - Kernel Bayes is also popular

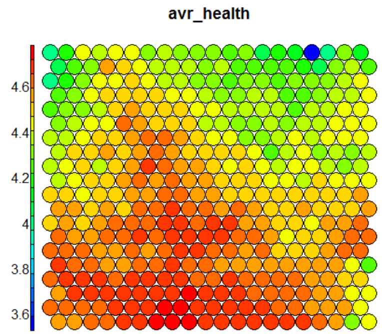
References:
http://www.cs.utah.edu/~piyush/teaching/15-9-print.pdf
http://web.stanford.edu/~hastie/Papers/svmtalk.pdf
http://sugiyama-www.cs.titech.ac.jp/ACML2010/ACML2010_fukumizu.pdf

Neural Network

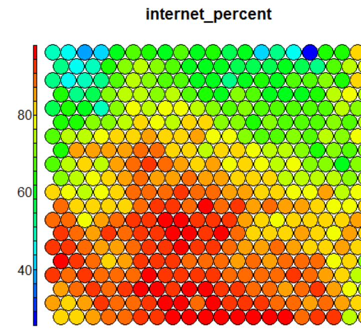
- Initial idea was from perceptron (Linear!!!)
- Neural network (Hidden layers make non-linearity possible)
- Trained using Back-propagation
  - How many hidden layers:
    - Geometric pyramid: sqrt(input*output)
    - Baum Haussler: $N_{hidden} < \frac{N_{hidden} * E_{tolerance}}{N_{input} N_{output}}$
    - Num weights H*(I + O)+H+O < 1/100 * training samples
  - Data imbalance: Jittering, Smoting
  - Stopping criteria: Increase in validation error (overfit)
  - Adjustment parameters: Momentum, learning rate

Neural Network

- Salient points to note after training:
  - Check sensitivity on confusing inputs
    - If the original problem is multi-class, club confusing classes together. Train a different 1-1 classifier that will be used for these classes alone
  - Identify threshold scores
  - Examine accuracy on held-out data
  - Vanishing gradients is a problem



- SOM:
  - Unsupervised
  - Learn weights from input vectors to nodes
  - Plot different attributes on a heat map
  - Terminating condition: No change in weights



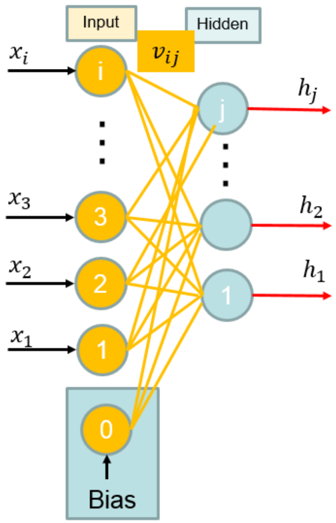
Neural Networks

- Autoencoders:
  - Set output same as input, add noise to input while learning
  - Stack to create deep neural nets
  - Use sparsity to ensure few connections/features per class

$$sparsity_j = \sum_{j=1}^{j} p \log \frac{p}{\widehat{p}_j} + (1-p) log \frac{1-p}{1-\widehat{p}_j}$$

$$w_{jk} \leftarrow w_{jk} + \eta * \delta\_input_k * h_j + \beta * sparsity_j$$

- Restricted Boltzman Machines
  - Only one set of weights to learn
  - Dropout – random noise added to all nodes



K-Nearest Neighbor / Instance Based Learning

- Select "K" nearest from training data using Euclidian distance
  - Vote for classification, average for regression
  - How to select "K":
    - Theory: As K increases, accuracy should increase
    - Real life: Try different K on hold out data
  - Wilson editing / condensation: Prune
- In theory, the best classifier
  - Maximum error for a good "K" should be 2*Bayes error

http://www.cs.haifa.ac.il/~rita/ml_course/lectures/KNN.pdf

K-Nearest Neighbor / Instance Based Learning

- Can be used to decide "goodness" of data
- Try K-NN, measure accuracy
- If a target classifier is worse than K-NN:
  - Classifier is not suitable for the problem
  - Data is noisy

Collaborative Filtering

- $\hat{R}_{ik} = \bar{R}_i + \alpha \sum_{X_j \in N_i} W_{ij}(R_{jk} - \bar{R}_j)$
  - $\hat{R}_{ik}: Rating\ of\ user\ i\ on\ movie\ k$
  - $\bar{R}_i: Average\ rating\ of\ user\ i$
  - $\alpha: \left(\sum |W_{ij}|\right)^{-1}, N_i: All\ users$
  - $W_{ij} = \frac{\sum_k (R_{ik} - \bar{R}_i)(R_{jk} - \bar{R}_j)}{\sqrt{\sum_k (R_{ik} - \bar{R}_i)^2 (R_{jk} - \bar{R}_j)^2}}$ Similarity or pearson coefficient
- Use normalized ratings

Radial Basis Functions

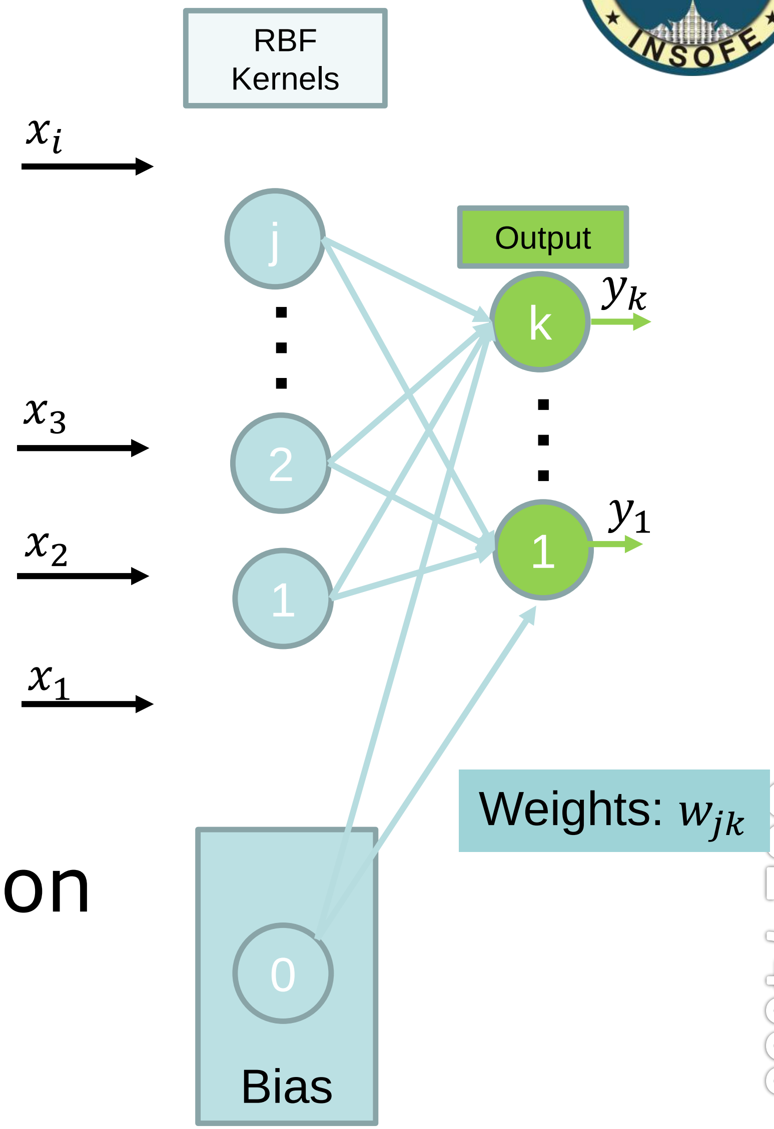
RBF network output:

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

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

$$g: kernel\ function$$

- A form of "Stacking"
  - "g" can be a weak classifier
    - K-means cluster center
    - Gaussian
  - Train w_{jk} using back-propagation



RBF network

Decision Tree

- Explainable results
- Can work with small data
  - Hubble data: 20 attributes, 2K samples
- Sensitive to noise, but:
  - Early stopping will help
- Information gain ratio good enough for most problems
  - Can add user specific weights to cost
  - Support: num of times rule occurs / number of samples
  - Confidence: Count of consequent and antecedent occurring together / count of antecedent
  - Lift: Confidence / Probability of consequent

SVM

- Suitable for sparse data if:
  - Noise is less (Check first with KNN/KMeans)
  - Linearly separable in Kernel space
  - Depends only on supports and can give theoretically optimal solution
- Multi-class SVM can be done by:
  - One versus Many: Number of classifiers equals number of classes (n)
  - One versus One:
    Number of classifiers: nC_2 $n: number\ of\ classes$

HMM

- Can model transitions and attributes simultaneously
  - E.g: Business or software processes, user behavior, NLP, Time series, hardware errors
- Handles sparse data with known structure
- Fast re-training possible: Only transitions and emissions have to be updated
- Can have one model for each pattern

Ensemble Methods

- Stacking
  - Learn several classifiers, use these outputs in a different classifier
- Bagging
  - Design several classifiers by taking random data splits, random attribute splits
  - Random Forests: Highly accurate
- Boosting
  - Learn –ve samples
  - Can help decide if data is very noisy or does not meet classifier assumptions (non-linear)

Inspire...Educate...Transform.

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

Appendix

WORKED-OUT EXAMPLE: RULE GENERATION

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

If ? then recommendation = hard

For the unknown term ?, we have nine choices:

age = young 2/8

age = pre-presbyopic 1/8

age = presbyopic 1/8

spectacle prescription = myope 3/12

spectacle prescription = hypermetrope 1/12

astigmatism = no 0/12

astigmatism = yes 4/12

tear production rate = reduced 0/12

tear production rate = normal 4/12

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

If astigmatism = yes and ? then recommendation = hard

age = young 2/4
age = pre-presbyopic 1/4
age = presbyopic 1/4
spectacle prescription = myope 3/6
spectacle prescription = hypermetrope 1/6
tear production rate = reduced 0/6
tear production rate = normal 4/6

If astigmatism = yes and tear production rate = normal and ? then recommendation = hard

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
young	myope	yes	normal	hard
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	yes	normal	none

age = young 2/2
age = pre-presbyopic 1/2
age = presbyopic 1/2
spectacle prescription = myope 3/3
spectacle prescription = hypermetrope 1/3

If astigmatism = yes and tear production rate = normal and spectacle prescription = myope then recommendation = hard

All rules from the set

```
IF    TearProduction = reduced
THEN  ContactLenses = none       [#soft=0 #hard=0 #none=12]

IF    TearProduction = normal
  AND Astigmatism = no
THEN  ContactLenses = soft       [#soft=5 #hard=0 #none=1]

IF    TearProduction = normal
  AND Astigmatism = yes
  AND SpectaclePrescription = myope
THEN  ContactLenses = hard       [#soft=0 #hard=3 #none=0]

IF    TearProduction = normal
  AND Astigmatism = yes
  AND SpectaclePrescription = hypermetrope
THEN  ContactLenses = none       [#soft=0 #hard=1 #none=2]
```

Table 1.2. Classification rules induced from the contact lens dataset. The number of covered examples of each class is also given.