Predictive Analytics

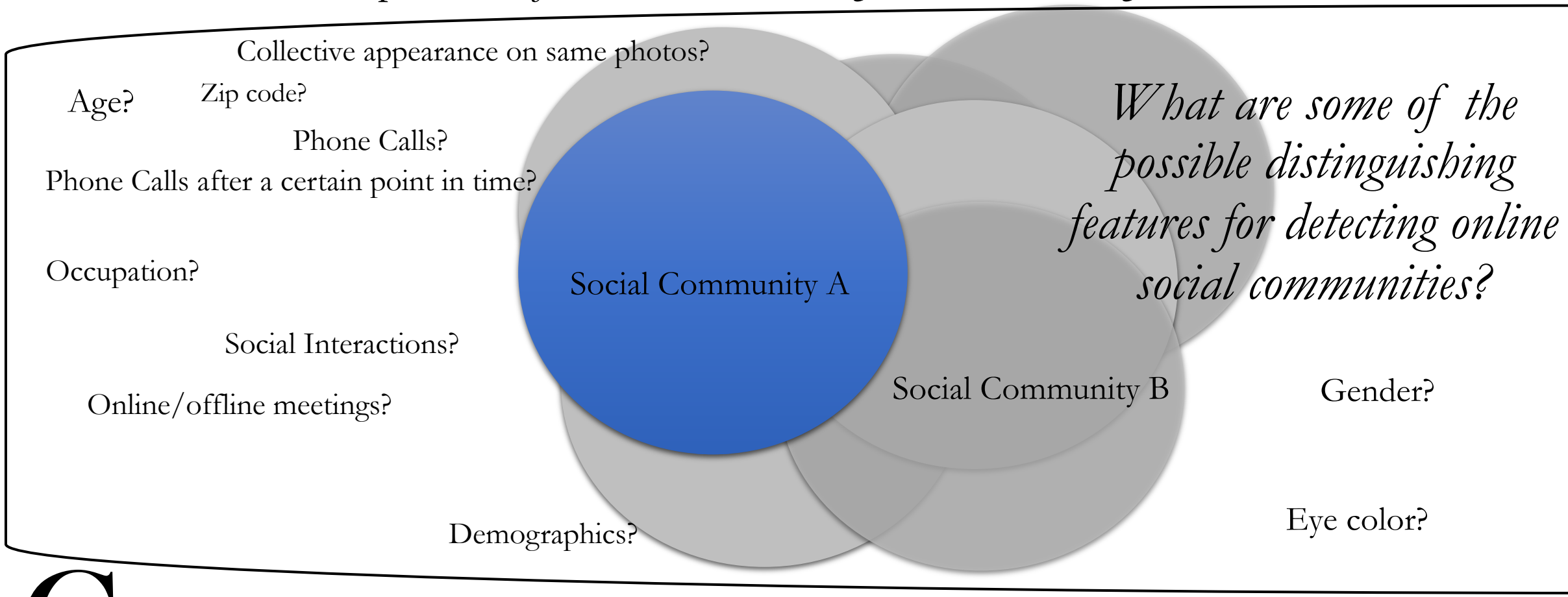
Chapter Six

Feature Selection Algorithms

"Before anything else, preparation is the key to success."
Alexander Graham Bell

Anasse Bari, Ph.D.

Consider the research problem *of* ***Online Community Detection in Dynamic Social Networks***



Community detection has been a research problem at the intersection of machine learning and social networks analytics. Community detection algorithms can allow us to uncover organizational principles in networks. When detecting communities, there are two possible sources of information one can use: the network structure, and the features and attributes of the data records (network nodes).

Learning Outcomes

- Understanding the process of selecting and extracting features
- Learning how to measure the predictive power of features
- Learning greedy algorithms for selecting the near-optimal set of features
- Learning the difference between feature selection and feature extraction
- Learning entropy based measure for ranking features
- Gaining hands-on practice on R and Weka

Outline

- Introduction to Feature Selection
- Feature Selection Algorithms
  - Feature Subset Selection
  - Feature Ranking
- The different between Feature Selection and Feature Extraction
- Feature Extraction Algorithms
  - Application: Introduction to Text Mining

Introduction

- Feature Selection known as ***subset selection*** is pre-step to applying models.
- The subset is of the features ***available from the dataset.***
- The best subset contains the ***least number of dimensions that most contribute to the accuracy of the model.***
- The remaining are discarded (are unimportant).
- ***This is a very important step in Data Analytics***
- The most useful part of this phase is attribute selection (also called feature selection)
  - Select relevant attributes
  - Remove ***redundant (e.g X = 2Y+1***) and/or irrelevant attributes – X represents values from column one, and Y represent values from column two.

Feature Selection vs Dimensionality Reduction

- **Dimensionality Reduction**
  - **Reduces the dimensionality without necessarily preserving the units/the actual attributes (e.g. PCA..)**
  - Dimensionality reduction can be a data transformation or data reduction
  - The measurement units of the features are not preserved, **you will have new features (in most cases – e.g. PCA: PCA1, PCA2…)**
- **Feature Selection**
  - You may only need few feature to perform analytics (e.g. Classification, Clustering…)
  - **The measurement units (length, weight, etc.) of the features are preserved.** Same Features (just lower number of the original set of features)

Reasons for Attribute (Feature) Selection

- **Simpler model**
  - More transparent
  - Easier to interpret
- **Faster model induction**
  - What about overall time? For example, for classification a problem, less data means that algorithms train faster.
- **Structural knowledge**
  - Knowing which attributes are important may be inherently important to the application
- What about the accuracy and Overfitting?
  - **Reduces Overfitting:** Less redundant data means ***less*** opportunity to make decisions based on noise.
  - Improves Accuracy: ***Less misleading*** data means modeling accuracy improves.

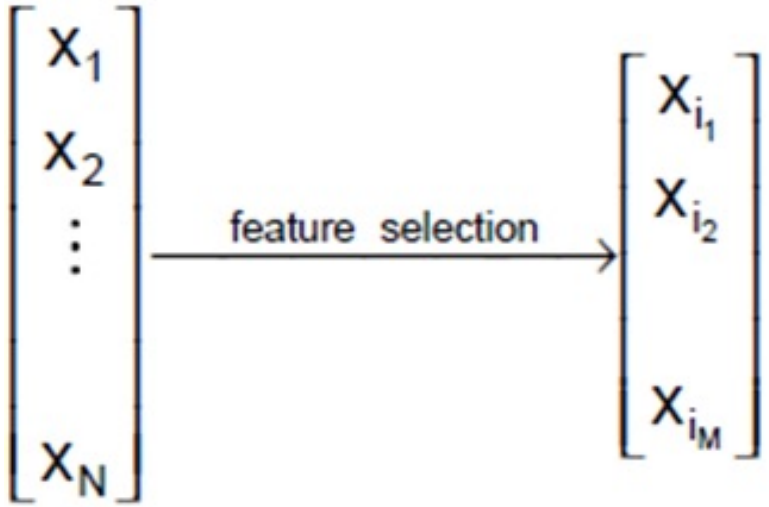
Overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably"

Feature Subset Selection (FSS)

Definition

Given a feature set $X=\{x_i \mid i=1\ldots N\}$

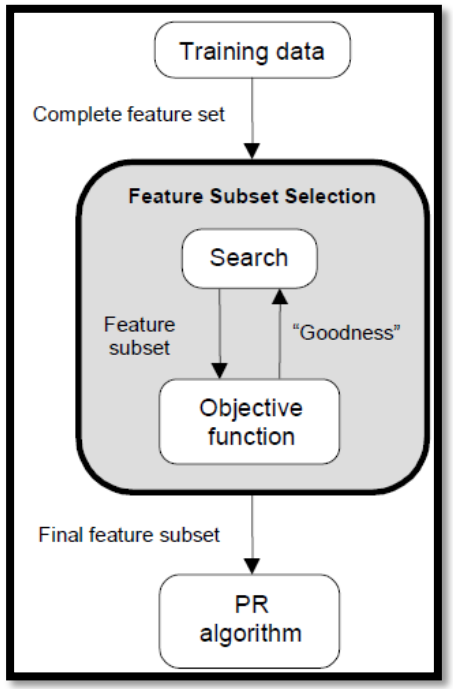
find a subset $YM=\{x_{i1}, x_{i2}, \ldots, x_{iM}\}$, with $M<N$, that optimizes ***an objective function J(Y), ideally the probability of correct classification.***



$$\{x_{i_1}, x_{i_2}, \ldots, x_{i_M}\} = \underset{M, i_m}{\operatorname{argmax}}[\{x_i \mid i=1\ldots N\}]$$

Feature Selection: Search strategy & an objective function

- Feature Subset Selection requires
  - A **search strategy** to select candidate subsets
  - An **objective function** to evaluate these candidates features



Reference to Weka (on the practice to Weka Feature Selection):

- Attribute Evaluator: Method by which attribute subsets are assessed.
- Search Method: Method by which the space of possible subsets is searched.

Feature Selection: Search strategy & an objective function

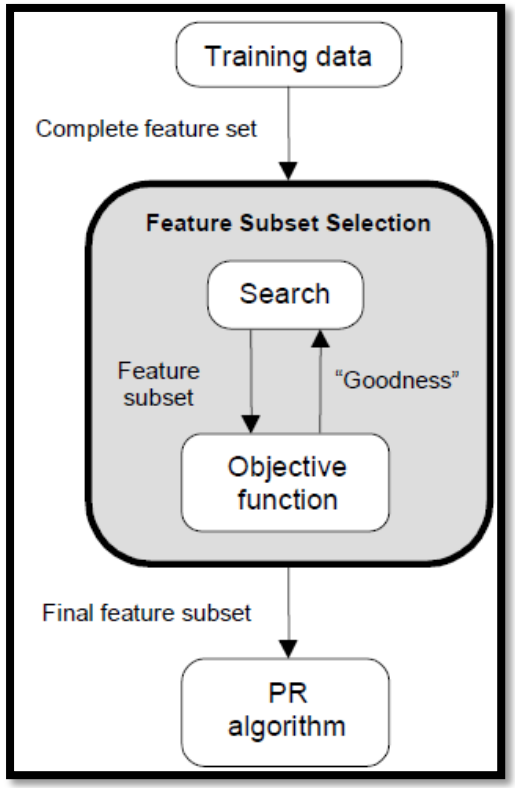
▪ Search Strategy

- Exhaustive evaluation of feature subsets involves $\binom{N}{M}$ *(N the number of feature, M is the number of the reduced number of features)*

  combinations for a fixed value of M, and 2^N combinations if M must be optimized as well

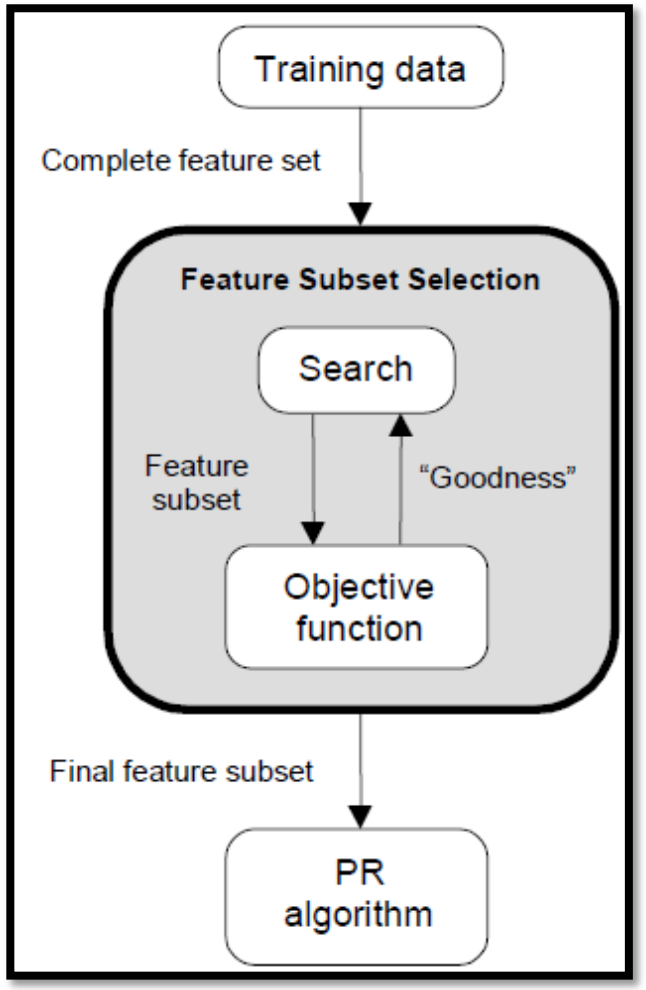
  - **If N is 100 and M is 10. Then to select 10 from 100, the number of possible combination is great than 2pow100.. (Brute Force)**

- This number of combinations is **unfeasible**, even for moderate values of M and N, so a search procedure must be used in practice
- For example, exhaustive evaluation of 10 out of 20 features involves 184,756 feature subsets; exhaustive evaluation of 10 out of 20 involves more than 1013 feature subsets
- A search strategy is therefore needed to direct the FSS (Feature Subset Selection) process as it explores the space of all possible combination of features



Search strategy and objective function

- **Objective Function**
  - The objective function evaluates candidate subsets and returns a measure of their "goodness", a ***feedback*** that is used by the search strategy to select new candidates

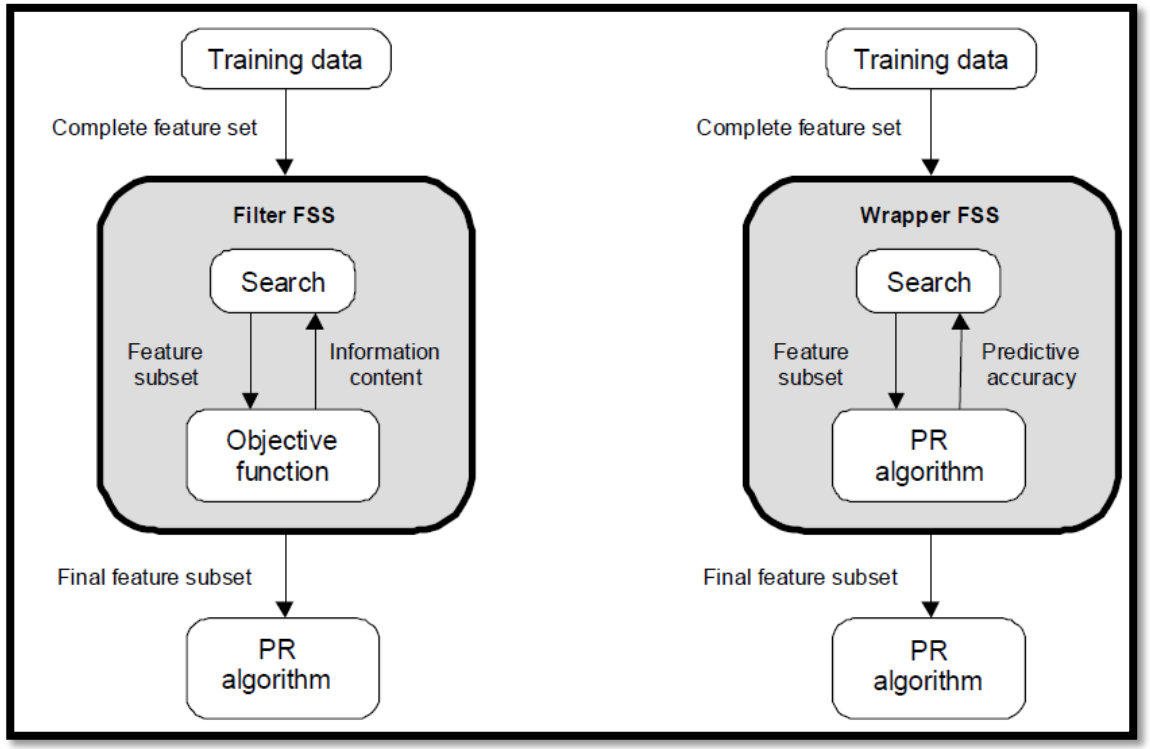


(in linear programming) An objective function is the function that it is desired to maximize or minimize.

Objective function

Objective functions are divided in two groups

- **Filters:** The objective function evaluates feature subsets by their ***information content***, typically ***interclass distance***, ***statistical dependence*** or information-theoretic measures. **Correlation between features and the class.**
- **Wrappers:** The objective function is a classifier, which evaluates feature subsets by their predictive accuracy (recognition rate on test data) by statistically resampling or cross-validation. ***Assess subsets using a classifier that you specify.***



Filter Types

- Distance or separability measures (Similarity Measure)

  These methods use distance metrics to measure ***class separability***, such as:

  Distance between classes: Euclidean, Mahalanobis, etc. (Review the chapter on similarity measures)

Filter Types

- These methods are based on the rationale that good feature subsets contain features highly correlated with (predictive of) ***the class, yet uncorrelated with (not predictive of) each other***
- Linear relation measures:

  Linear relationship between variables can be measured using the **correlation coefficient**

$$J(Y_M) = \frac{\sum_{i=1}^{M} P_{ic}}{\sum_{i=1}^{M} \sum_{j=i+1}^{M} Pij}$$

Where P_{ic} is the correlation coefficient between feature 'i' and the class label and P_{ij} is the correlation coefficient between features 'i' and 'j'

> **"Correlation coefficients measure the strength of association between two variables. The most common correlation coefficient, called the Pearson product-moment correlation coefficient, measures the strength of the linear association between variables."** to refresh on Correlation Coefficients Read: http://stattrek.com/statistics/correlation.aspx?Tutorial=AP

Non-Linear relation measures: Correlation is only capable of measuring linear dependence. A more powerful method is the **mutual information. [2]**

Formally, the mutual information of two discrete random variables X and Y can be defined as:[1]:20

$$\mathrm{I}(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x,y) \log \left(\frac{p(x,y)}{p(x)\, p(y)} \right),$$

where $p(x,y)$ is the joint probability function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

Filters vs. Wrappers

- **Filters (based on correlation or distance btw features and the class)**
  - Advantages
    - ***Fast execution:*** Filters generally involve a non-iterative computation on the dataset, which can execute much faster than a classifier training session
    - **Generality: Since filters evaluate the intrinsic properties of the data, rather than their interactions with a particular classifier, their results exhibit more generality: the solution will be "good" for a larger family of classifiers**
  - Disadvantages
    - Tendency to select large subsets: Since the filter objective functions are generally monotonic, the filter tends to select the full feature set as the optimal solution. This forces the user to select an arbitrary cutoff on the number of features to be selected

Filters vs. Wrappers

- **Wrappers (based on Classifier)**
  - Advantages
    - Accuracy: wrappers generally achieve better recognition rates than ***filters*** since they are tuned to the specific interactions between the classifier and the dataset.
    - Ability to generalize: wrappers have a mechanism to avoid overfitting, since they typically use cross-validation measures of predictive accuracy.
  - Disadvantages
    - Slow execution: since the wrapper must train a classifier for each feature subset (or several classifiers if cross-validation is used), the method can become unfeasible for computationally intensive methods
    - Lack of generality: the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function. The "optimal" feature subset will be specific to the classifier under consideration

Examples from Weka
Attribute Evaluator

CfsSubsetEval: Values subsets that correlate highly with the class value and low correlation with each other.

ClassifierSubsetEval: Assesses subsets using a predictive algorithm and another dataset that you specify.

WrapperSubsetEval: Assess subsets using a classifier that you specify and n-fold cross validation.

Weka

Feature Candidate's Search Strategies

- The Search Method is the structured way in which the search space of possible attribute subsets is navigated based on the subset evaluation (pervious slide)
- There exists a large number of search strategies that can be grouped in three categories:
  - **Exponential algorithms (known as brute-force search, exhaustive search)**
    - These algorithms evaluate a number of subsets that grow exponentially with the dimensionality of the search space.
  - **Sequential algorithms:**
    - These algorithms add or remove features sequentially, but have a tendency to become trapped in local minima. Representative examples of sequential search include:
      - **Sequential Forward Selection**
      - Sequential Backward Selection
      - Plus-l Minus-r Selection
      - Bidirectional Search
      - Sequential Floating Selection

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Reprinted by permission from IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS
Vol. SMC-4, No. 1, January 1974, pp. 116-117

T.M. Cover: "The Best Two Independent Measurements are Not the Two Best," ...

The Best Two Independent Measurements Are Not the Two Best

THOMAS M. COVER

***Abstract*—Consider an item that belongs to one of two classes, $\theta = 0$ or $\theta = 1$, with equal probability. Suppose also that there are two measurement experiments E_1 and E_2 that can be performed, and suppose that the outcomes are independent (given θ). Let E_i' denote an independent performance of experiment E_i. Let $P_e(E)$ denote the probability of error resulting from the performance of experiment E. Elashoff [1] gives an example of three experiments E_1,E_2,E_3 such that $P_e(E_1) < P_e(E_2) < P_e(E_3)$, but $P_e(E_1,E_3) < P_e(E_1,E_2)$. Toussaint [2] exhibits binary valued experiments satisfying $P_e(E_1) < P_e(E_2) < P_e(E_3)$, such that $P_e(E_2,E_3) < P_e(E_1,E_3) < P_e(E_1,E_2)$. We shall give an example of binary valued experiments E_1 and E_2 such that $P_e(E_1) < P_e(E_2)$, but $P_e(E_2,E_2') < P_e(E_1,E_2) < P_e(E_1,E_1')$. Thus if one observation is allowed, E_1 is the best experiment. If two observations are allowed, then two independent copies of the "worst" experiment E_2 are preferred. This is true despite the conditional independence of the observations.**

The Bayes probability of error is given for a discrete random variable X by

$$P_e(E) = \sum_x \min \{\Pr \{\theta = 0\}P_0(x), \Pr \{\theta = 1\}P_1(x)\}.$$

Thus, for example,

$$P_e(E_1) = \tfrac{1}{2} \min \{1 - p_0, 1 - p_1\} + \tfrac{1}{2} \min \{p_0, p_1\}$$
$$= \tfrac{1}{2}[1 - |p_0 - p_1|].$$

Choose

$$p_0 = 0.96, p_1 = 0.04, r_0 = 0.9, r_1 = 0.$$

We then have

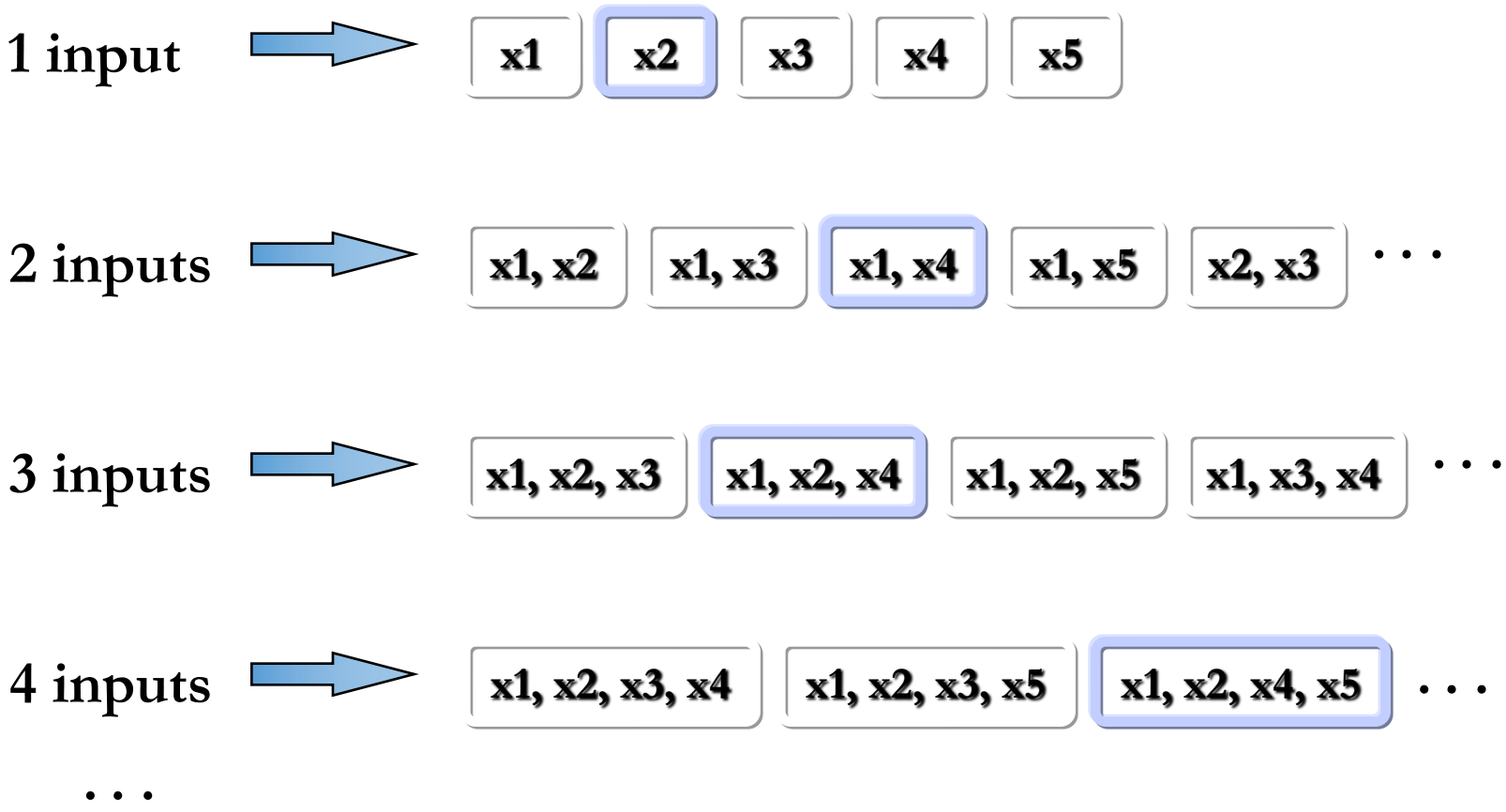
$$P_e(E_1) = 0.04$$
$$< P_e(E_2) = 0.05$$

and

$$P_e(E_2,E_2') = 0.005$$
$$< P_e(E_1,E_2) = 0.022$$

Exhaustive Search Example

Direct exhaustive search



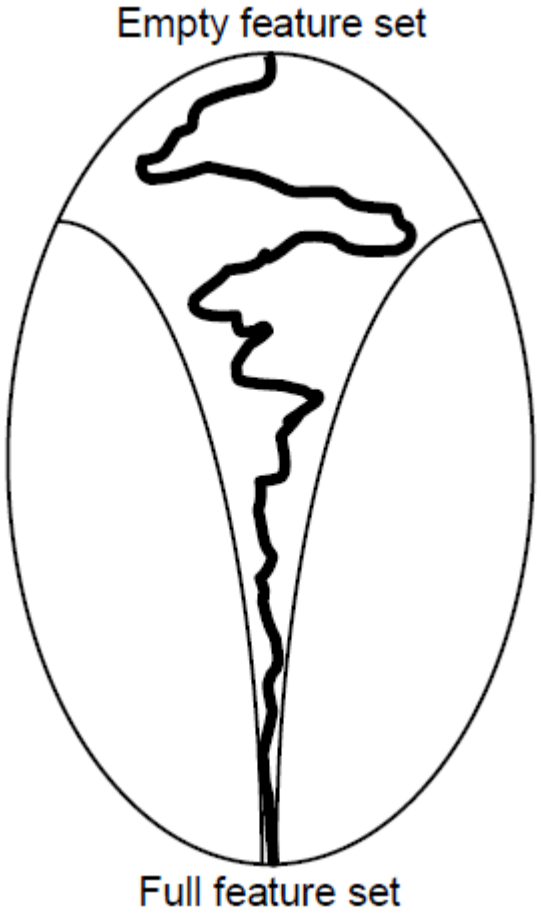
Source: mirlab.org/jang/books/dcpr/slide/featureSelect.ppt

Sequential Forward Selection Algorithm (SFS)

- Sequential Forward Selection is ***the simplest greedy search algorithm***
- Starting from the empty set, sequentially add the feature x^+ that results in the highest value of the objective function $J(Y_k+x^+)$ when combined with the features Y_k that have already been selected
- Algorithm:

1. Start with the empty set $Y=\{\varnothing\}$
2. Select the next best feature $x^+ = \underset{x \in X-Y_k}{\operatorname{argmax}}[J(Y_k + x)]$
3. Update $Y_{k+1}=Y_k+x$; $k=k+1$
4. Go to 2

Source: http://www.facweb.iitkgp.ernet.in/~sudeshna/courses/ML06/featsel.pdf



Sequential Forward Selection Algorithm (SFS)

1. Shuffle the data set and split into a training set of 70% of the data and a testset of the remaining 30%.
2. Let j vary among feature-set sizes: $j = (0, 1, 2, \ldots, m)$
   a. Let fs_j = best feature set of size j, where "best" is measured as the minimizer of the leave-one-out cross-validation error over the training set.
   b. Let $Testscore_j$ = the RMS prediction error of feature set fs_j on the test set.

   End of loop of *(j)*.
3. Select the feature set fs_j for which the test-set score is minimized.

Leave-one-out cross-validation
Leave-*one*-out cross-validation (**LOOCV**) is a particular case of leave-*p*-out cross-validation with $p = 1$.

Cascaded cross-validation procedure for finding the best set of up to m features.
RMS: Root-mean-square error

Sequential Forward Selection (SFS)

More details on This is the "best" is being selected! (See previous slide – Highlighted in blue step

Full procedure for evaluating feature selection of up to m attributes.

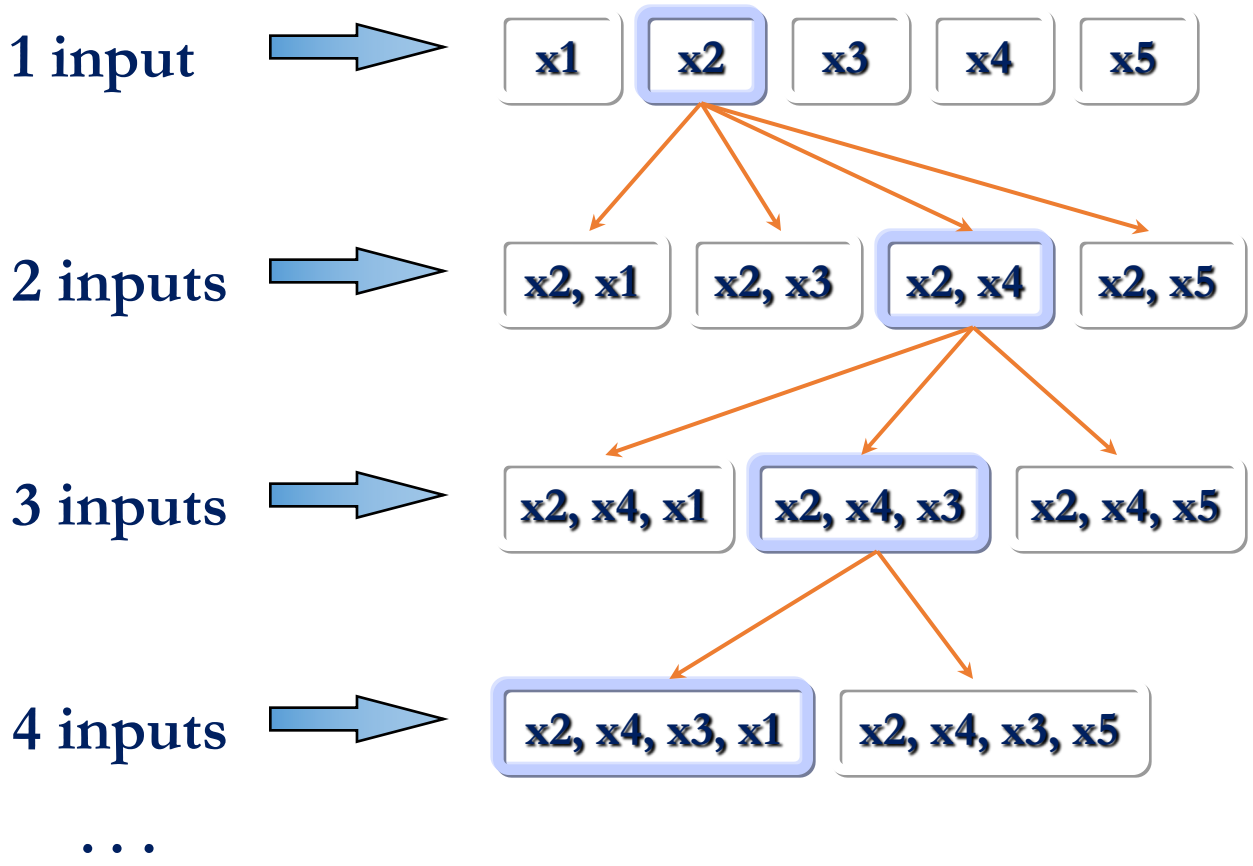
1. Collect a training data set from the specific domain.
2. Shuffle the data set.
3. Break it into *P* partitions, (say $P = 20$)
4. For each partition ($i = 0, 1, ..., P-1$)
   a. Let *OuterTrainset(i)* = all partitions except *i*.
   b. Let *OuterTestset(i)* = the *i*'th partition
   c. Let *InnerTrain(i)* = randomly chosen 70% of the *OuterTrainset(i)*.
   d. Let *InnerTest(i)* = the remaining 30% of the *OuterTrainset(i)*.
   e. For $j = 0, 1, ..., m$
      Search for the best feature set with *j* components, fs_{ij} using leave-one-out on *InnerTrain(i)*
      Let $InnerTestScore_{ij}$ = RMS score of fs_{ij} on *InnerTest(i)*.
      End loop of *(j)*.
   f. Select the fs_{ij} with the best inner test score.
   g. Let $OuterScore_i$ = RMS score of the selected feature set on *OuterTestset(i)*
   End of loop of *(i)*.
5. Return the mean Outer Score.

Leave-one-out cross-validation

Leave-*one*-out cross-validation (**LOOCV**) is a particular case of leave-*p*-out cross-validation with $p = 1$.

Sequential Forward Selection

- Sequential forward selection (SFS)



Source: mirlab.org/jang/books/dcpr/slide/featureSelect.ppt

Sequential Forward Selection (SFS)

- SFS performs best when the optimal subset has a small number of features.
- When the search is near the empty set, a large number of states can be potentially evaluated.
- Towards the full set, the region examined by SFS is narrower since most of the features have already been selected.
- The search space is drawn like an ellipse to emphasize the fact that there are fewer states towards the full or empty sets.

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Some More Feature Subset Selection Algorithms

- Brute-force approach:
  - Try all possible feature subsets as input to data mining algorithm
- Embedded approaches:
  - Feature selection occurs naturally as part of the data mining algorithm
- LVF: Las Vegas Filter using consistency (cRate)
  - (1) randomly generate a subset S from the full set
  - if it satisfies prespecified cRate, keep S with min #S
  - go back to 1 until a stopping criterion is met
- Many other algorithms: SBS, B&B, ...

Source: Book Data Mining: Concepts and Techniques

Feature Selection by Ranking Features

Feature Selection by Ranking Features

- One may be tempted to evaluate each individual feature separately and select those M features with the highest scores. This often referred as Feature Ranking.
- There are some algorithms to rank (get the features score) the features:
  - **Entropy**
  - **Gain Ratio (Entropy Based)***
  - **Information Gain (Entropy Based)***
  - **Chi-square***
  - **SVM Ranker***
- Unfortunately, this strategy might not always work since it does not account for feature dependence. ***However, it can be done as a combined effort with Forward Selection Algorithm (the approach most adopted in practice)***

Reprinted by permission from IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS
Vol. SMC-4, No. 1, January 1974, pp. 116-117

T.M. Cover: "The Best Two Independent Measurements are Not the Two Best," ...

The Best Two Independent Measurements Are Not the Two Best

THOMAS M. COVER

***Abstract*—Consider an item that belongs to one of two classes, $\theta = 0$ or $\theta = 1$, with equal probability. Suppose also that there are two measurement experiments E_1 and E_2 that can be performed, and suppose that the outcomes are independent (given θ). Let E_i' denote an independent performance of experiment E_i. Let $P_e(E)$ denote the probability of error resulting from the performance of experiment E. Elashoff [1] gives an example of three experiments E_1,E_2,E_3 such that $P_e(E_1) < P_e(E_2) < P_e(E_3)$, but $P_e(E_1,E_3) < P_e(E_1,E_2)$. Toussaint [2] exhibits binary valued experiments satisfying $P_e(E_1) < P_e(E_2) < P_e(E_3)$, such that $P_e(E_2,E_3) < P_e(E_1,E_3) < P_e(E_1,E_2)$. We shall give an example of binary valued experiments E_1 and E_2 such that $P_e(E_1) < P_e(E_2)$, but $P_e(E_2,E_2') < P_e(E_1,E_2) < P_e(E_1,E_1')$. Thus if one observation is allowed, E_1 is the best experiment. If two observations are allowed, then two independent copies of the "worst" experiment E_2 are preferred. This is true despite the conditional independence of the observations.**

The Bayes probability of error is given for a discrete random variable X by

$$P_e(E) = \sum_x \min \{\Pr \{\theta = 0\}P_0(x), \Pr \{\theta = 1\}P_1(x)\}.$$

Thus, for example,

$$P_e(E_1) = \tfrac{1}{2} \min \{1 - p_0, 1 - p_1\} + \tfrac{1}{2} \min \{p_0, p_1\}$$
$$= \tfrac{1}{2}[1 - |p_0 - p_1|].$$

Choose

$$p_0 = 0.96, p_1 = 0.04, r_0 = 0.9, r_1 = 0.$$

We then have

$$P_e(E_1) = 0.04$$
$$< P_e(E_2) = 0.05$$

and

$$P_e(E_2,E_2') = 0.005$$
$$< P_e(E_1,E_2) = 0.022$$

Algorithms that can help you select the best Features (Feature Ranking)

- Information Gain (Entropy Based)
- Gain Ratio (Entropy Based)
- Fisher Score (F-score)
- Chi-square
- SVM Ranker

Entropy

- In information theory, Entropy measures ***the amount of information*** in a random variable; it's the average length of the message needed to transmit an outcome of that variable using the optimal code.
  - Uncertainty, Surprise, Information
  - *"High Entropy"* means X is from a uniform (boring) distribution
  - *"Low Entropy"* means X is from a varied (peaks and valleys) distribution

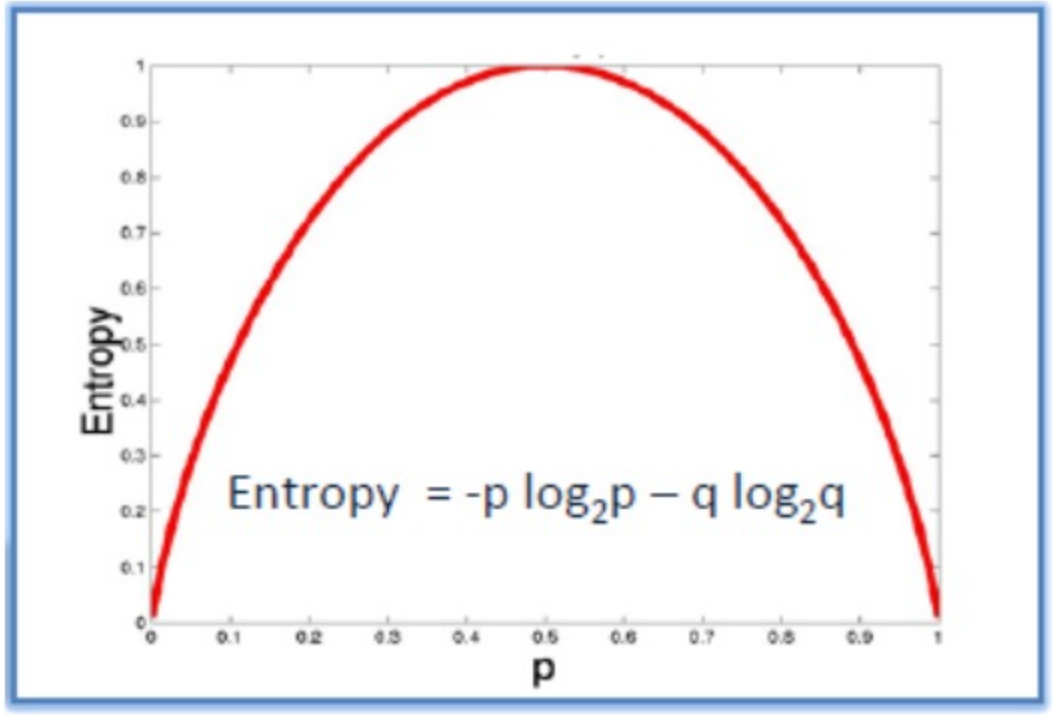
- Entropy-based measure for attribute selection is used in Decision Trees construction (to be elaborated more in the data classification chapter)

Information Gain

$$Entropy = \sum_{d \in Decisions} -p(d) * \log(p(d))$$

$$InformationGain(A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

S: the predictive column
A: Attribute
See detailed example in slide 35, 36, 37, 38.



Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

Definition of Entropy

- Entropy $$H(X) = \sum_{x \in A_X} -P(x) \log_2 P(x)$$
- Conditional Entropy:

$$H(X \mid Y) = \sum_{y \in A_Y} P(y) H(X \mid y)$$

Example: Let's Play Tennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example from Book "Data Mining: Practical Machine Learning Tools and Techniques"

Calculating the Entropy of Attributes

- We want to make decisions based on one of the attributes
- We will rank the attributes based on their prediction power
- There are four attributes to choose from:
  - **Outlook**
  - **Temperature**
  - **Humidity**
  - **Wind**

To decide whether to play Tennis or not

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example

What is Entropy of play tennis?

Entropy of Play = -5/14*log2(5/14) – 9/14*log2(9/14)
= Entropy(5/14,9/14) = 0.9403

- Now based on Outlook, divided the set into three subsets, compute the entropy for each subset
- The expected conditional entropy is:
  5/14 * Entropy(3/5,2/5) +
  4/14 * Entropy(1,0) +
  5/14 * Entropy(3/5,2/5) = 0.6935

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Outlook Continued

- The expected conditional entropy is:
  5/14 * Entropy(3/5,2/5) +
  4/14 * Entropy(1,0) +
  5/14 * Entropy(3/5,2/5) = 0.6935
- So IG(Outlook) (information gain) = 0.9403 – 0.6935 = 0.2468 (see more details on the next slide)
- We seek an attribute that makes partitions as pure as possible

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Temperature

- Now let us look at the attribute Temperature
- The expected conditional entropy is:
  4/14 * Entropy(2/4,2/4) +
  6/14 * Entropy(4/6,2/6) +
  4/14 * Entropy(3/4,1/4) = 0.9111
- So IG(Temperature) = 0.9403 – 0.9111 = 0.0292

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Humidity

- Now let us look at attribute Humidity
- What is the expected conditional entropy?
- 7/14 * Entropy(4/7,3/7) +
  7/14 * Entropy(6/7,1/7) = 0.7885
- So IG(Humidity) = 0.9403 – 0.7885
  = 0.1518

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Wind

- What is the information gain for wind?
- Expected conditional entropy:
  8/14 * Entropy(6/8,2/8) +
  6/14 * Entropy(3/6,3/6) = 0.8922
- IG(Wind) [information gain] = 0.9403 – 0.8922 = 0.048

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Information Gains

- Outlook 0.2468
- Temperature 0.0292
- Humidity 0.1518
- Wind 0.0481
- We choose Outlook since it has the highest information gain

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Fisher Score (F-score)

Fisher Score [10]: Features with high quality should assign similar values to instances in the same class and different values to instances from different classes. With this intuition, the score for the i-th feature S_i will be calculated by Fisher Score as,

$$S_i = \frac{\sum_{k=1}^{K} n_j(\mu_{ij} - \mu_i)^2}{\sum_{k=1}^{K} n_j \rho_{ij}^2}, \tag{0.2}$$

where μ_{ij} and ρ_{ij} are the mean and the variance of the i-th feature in the j-th class respectively, n_j is the number of instances in the j-th class, and μ_i is the mean of the i-th feature.

Jiliang Tang, Salem Alelyani and Huan Liu
Feature Selection for Classification: A Review

Chi-square (pronounced kiy—square)

- Another popular feature selection method is $\chi 2$.
- In statistics, the $\chi 2$ test is applied to test the independence of two events, where two events A and B are defined to be independent if
  - $P(AB) = P(A)P(B)$
  - or, equivalently, $P(A/B) = P(A)$ and $P(B/A) = P(B)$
- In feature selection, the two events are occurrence of the term and occurrence of the class.
- $\chi 2$ is a measure of how much expected counts E and observed counts N deviate from each other.
- If the two events are dependent, then the occurrence of the term makes the occurrence of the class more likely (or less likely), so it should be helpful as a feature. This is the rationale of $\chi 2$ feature selection.

For more details on how to calculate the Chi-square: http://math.hws.edu/javamath/ryan/ChiSquare.html

SVM Ranker

- Create a linear SVM model for all features. *(Support Vector Machines will be covered in details in the Data Classification Chapter)*
- The calculated weights (ω) can be used to decide the relevance of each feature
- The larger $|\omega_j|$ is, the j^{th} feature plays a more important role in the decision function
- Only ω in linear SVM model has this indication, so
  - This approach is restricted to linear SVM.
- We thus rank features according to $|\omega_j|$.

Source Book: Data Mining: Practical Machine Learning Tools and Techniques

Feature Extraction

“

Re-visiting: Difference between Feature Selection and Feature Extraction

- We can see Feature Extraction from two different aspects:
  - Feature extraction consists in transforming arbitrary **data**, such as text or images, into numerical features usable for machine learning such as TF or TFIDF.
  - Feature extraction is transforming the **existing features** into a lower dimensional space such as PCA (Principal component analysis). Note that the new dimensions are very different from the initial features.

Text Mining – Feature Extraction

Introduction and application to Feature Extraction

“

Vector Space Model (VSM)

- Representing textual data in a simple data model
- Documents as Vectors in an n-dimentional Euclidian space.
- The first step in modeling the document into a vector space is to create a dictionary of terms present in documents.
- To do that, you can simple select all terms from the document and convert it to a dimension in the vector space, but we know that there are some kind of words (stop words) that are present in almost all documents, and what we're doing is extracting important features from documents, features do identify them among other similar documents, so using terms like "the, is, at, on", etc.. isn't going to help us, so in the information extraction, we'll just ignore them.
- Words like "the,is,at,on" etc … are called stop words.

Vector Space Model (VSM)

- Let's define some statistics for text documents:
  - TF: term frequency
    - In the case of the term frequency tf(t,d), the simplest choice is to use the raw frequency of a term in a document, i.e. the number of times that term t occurs in document d.
  - IDF: Inverse document frequency
    - The inverse document frequency is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

    - N is the total number of documents in the corpus.
    - The denominator of above equation is the number of documents where the term t appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to

$$1 + |\{d \in D : t \in d\}|$$

Vector Space Model (VSM)

- Let's define some statistics for text documents:
  - TFIDF: Term frequency–Inverse document frequency
  - Then tf–idf is calculated as

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

  - A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents;
  - Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0.
  - As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

Vector Space Model (VSM)

- Let's take the documents below to define our document space:
  - Document 1:
    - d1: The sky is blue.
    - d2: The sun is bright.
  - Document 2:
    - d1: The sun in the sky is bright.
    - d2: We can see the shining sun, the bright sun.
  - Document 3:
    - d2: You can see the sun in the sky.

Dictionary	IDF
Sky	0
Sun	0
Blue	0.47
Bright	0.18
see	0.18

Source: http://blog.christianperone.com/2011/09/machine-learning-text-feature-extraction-tf-idf-part-i/

Vector Space Model (VSM)

- Let's extract the TF and TFIDF features of those documents:

TF					
Doc#	Sky	Sun	Blue	Bright	See
1	1	1	1	1	0
2	1	3	0	2	1
3	1	1	0	0	1

TFIDF					
Doc#	Sky	Sun	Blue	Bright	See
1	0	0	0.47	0.18	0
2	0	0	0	0.36	0.18
3	0	0	0	0	0.18

Dictionary	IDF
Sky	0
Sun	0
Blue	0.47
Bright	0.18
see	0.18

Source: http/christianperone.com/2011/09/

References & Resources

- Ricardo Gutierrez-Osuna , Pattern Recognition Slides, Wright State University
- Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." The Journal of Machine Learning Research 3 (2003): 1157-1182.
- Goodwin, Linda, et al. "Data mining issues and opportunities for building nursing knowledge." Journal of biomedical informatics 36.4 (2003): 379-388.
- George Bebis , Pattern Recognition Slides, University of Nevada, Reno
- https://en.wikipedia.org/wiki/Entropy_%28information_theory%29
- https://en.wikipedia.org/wiki/Information_theory
- https://users.cs.fiu.edu/~taoli , Data Mining: Concepts and Techniques Slides, Florida International University
- Chang, Yin-Wen, and Chih-Jen Lin. "Feature ranking using linear svm." Causation and Prediction Challenge Challenges in Machine Learning 2 (2008): 47.

References & Resources

- Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." The Journal of Machine Learning Research 3 (2003): 1157-1182.
- A.L. Yuille, Dimension Reduction & PCA slides, UCLA university
- https://en.wikipedia.org/wiki/Tf%E2%80%93idf
- http://www.tfidf.com/
- http://nlp.stanford.edu/IR-book/html/htmledition/feature-selectionchi2-feature-selection-1.html
- http://www.saedsayad.com/decision_tree.htm
- http://www.ling.upenn.edu/~clight/chisquared.htm