Unsupervised

Association Rule Mining

- Useful when launching a new product
- Sell more butter by making a promotion for bread
- Association can be used to group words for search engine algorithms: Lebron + NBA --> Lebron James

Association in Classification

- Imputing the response to one question based on the response of another to make smaller surveys

Clustering

- Requires a vector space
- Basketball: (Points, Assists, Rebounds) --> Clusters: (High Assist, Low Points); (High Everything); (Low Assistants, High Points)

Supervised

Least Squares Linear Regression

- Minimizes the sum of squared residuals (**SSQ**), also called the residual sum of squares (**RSS)**
  - Highly sensitive to outliers, which should be assessed for removal
- Mean squared error (**MSE**) = SSQ / |data points|

Residuals

- Counts by residual value should be normally distributed with mean zero
- Pattern-less across buckets of a variable

Consider a log transformation of a variable or piece-wise regression

1-Rule

Make a prediction based on the most likely response.

Ex:

	Front	Back
Male	2	20
Female	12	4

==> If Gender = M, then Front = No; If Gender = F then Front = Yes; **6 errors** (Males in the front + Females in the back)

	Front	Back
Device	3	11
No Device	11	13

==> If Device, then Front = No; If No Device then Front = No; **14 errors** (all people in front!)

*Find all rules. Choose feature with the **fewest total errors**.*

Naïve Bayes

$$P(A|\boldsymbol{B}) = \frac{P(A)P(\boldsymbol{B}|A)}{\sum P(E)P(B|E)}$$

$$Posterior = \frac{Prior \cdot Likelihood}{Evidence}$$

<u>Example</u>

Refund	Marital Status	-->	Cheat?

Algorithm

Apply Bayes Rule:

$$P(cheat \mid r, m) = \frac{P(cheat)\ P(\boldsymbol{r}, \boldsymbol{m} \mid cheat)}{P(cheat) * P(\boldsymbol{r},\ \boldsymbol{m} \mid cheat) + P(no\ cheat)P(\boldsymbol{r},\ \boldsymbol{m} \mid no\ cheat)}$$

Assume independence of feature variables:

$$\frac{P(cheat)\ P(\boldsymbol{r}|\ cheat)P(\boldsymbol{m}|\ cheat)}{P(cheat) * P(\boldsymbol{r} \mid cheat)P(\boldsymbol{m} \mid cheat) + P(no\ cheat) * P(\boldsymbol{r} \mid no\ cheat)P(\boldsymbol{m} \mid no\ cheat)}$$

Empirically derive.

Improvements

- Mitigate the issues with the independence assumption by appropriately clustering
- Although not desired, highly correlated variables need to be removed through feature engineering

<u>The loaded dice problem</u>

A casino has a fair dice 80% of the time and a loaded dice that rolls a one with a 50% chance.

$$P(Loaded|\ \{1,1,1,1\}) = \frac{P(Loaded)P(\{1,1,1,1\}|Loaded)}{P(\{1,1,1,1\})} = \frac{0.2\left(\frac{1}{2}\right)^4}{0.2\left(\frac{1}{2}\right)^4 + 0.8\left(\frac{1}{6}\right)^4}$$

Denominator: How can four consecutive ones be rolled?
Numerator: Specify one of way the outcome can happen.

Prism Rules

For each **feature**:

  For each distinct **value**:

    Generate a rule *if feature = value then outcome = <>*

Algorithm

- Start with single feature for a given outcome:
  - If outlook = sunny then play = **no** (3/5)
  - If outlook = rainy then play = **no** (3/7)
  - If outlook = mild then play = **no** (2/3)

  Numerator should add up to count of "no" *Denominator should add up to full set*

- Find the error of each feature-value rule
- Choose condition with the **highest % accuracy**
  - In case of a tie, choose the one with the **largest denominator**

- Now we have the best choice for the first condition
- We have to examine the remaining **ANDs**
  - Repeat using remaining features
  - Stop at perfect accuracy
    - If Outlook=Sunny and Humidity=Low then Play=Yes (3/3)

- If a rule set is done, delete the records corresponding to it and restart to cover all data points

Entropy Decision Tree Learning

Entropy

- Say X is 1 with probably p_1 and 0 with probability p_2

$$H(X) = -p_1 \ln(p_1) - p_2 \ln(p_2)$$

Observations:

- **Certainty**: H(X) = 0 if the outcome of X is certain
- **Patterns**: H(X) = ~~n * (1/n)~~ * ln(n) for n equal outcomes

Example

	Refund	**Marital Status**
Cheat	Split 1 to evaluate	Split 2 to evaluate
No Cheat		

	Refund = Yes ==> H(x) = 0	Refund = No ==> H(x) = 3/7ln(7/3)+4/7ln(7/4)
Cheat = Yes	0/3	3/7
Cheat = No	3/3	4/7

	Single ==> H(x) = ln(2)	Married ==> H(x) = 0	Divorced ==> ln(2)
Cheat = Yes	2/4	4/4	1/2
Cheat = No	2/4	0/4	1/2

At the root of the tree, we want the lowest entropy (as close to 0 as possible)

Use the weighted average entropy as a metric:

- Refund = (3/10)*0 + (7/10)...
- Marital status = (4/10)*ln(2) +(4/10)*0 + (2/10)*ln(2)

Comparison of Techniques

	Pros	Cons	Discussion
NB	- **Probability** of output - Uses **all features**	- Independence assumption - Must remove correlated features - Not human-readable!	
EDT	- Interpretable	- Can be **large** - Only the best feature used each iteration	Tree pruning (trimming bottom) to reduce size
1-R		- Too simple	Only the best feature has a voice
PRISM	- Every feature plays a part	- Model can be **large** - Over-fits (100% accuracy and coverage)	Create stopping rules to prevent over-fitting

Outlier Detection

- By-product of regression (residuals) and clustering (far from given cluster's center)
- Data must be understood before removed

Evaluation

Quadratic Loss

For each record:

$$\boldsymbol{QL} = \sum_{outcomes} [Probability - Actual]^2$$

- Ex: If record is M but model predicts: P(Y) = 0.2, P(N) = 0.45 P(M) = 0.35 --> QE = (**0**-0.2)^2+(**0**-0.45)^2+(**1**-0.35)^2

Confusion Matrix

- Deeply related to TP/TN and FP/FN table, but more detailed

	Predicted			
		A	B	C
Actual	A	TP_A		
	B		TP_B	
	C			TP_C

K-fold cross-validation

0. Set k (recommend k = 10)

1. Randomly split the labeled data into k folds
2. For each fold:
    a. Use the fold for **testing** and all other folds for **training**
    b. Record **e**
3. Find the average error

**If k = the size of the dataset, we get "leave one out" cross-validation (since only 1 record is used for testing)*

Challenges

- **Interpretation**
    - Stereotyping on only a subset of the population
- **Validation**
    - How to pick the best data mining technique?
- **Ethics**
- **Complexity**
    - Example: all combinations of models
- **Data skew**
    - Down-sampling (not enough 1s): removing 0s to make the proportion equal
    - Up-sampling (not enough 0s): keep all the 1s, but sample with replacement to get 0s

Variable Selection

- **Forward** – try each variable one by one and find the lowest **SSE** (not used in practice)
- **Backward** – try all the variables; remove the worst one (used more often)
- **Shrinkage** – LASSO: use matrix algebra to shrink coefficients to help eliminate variables

Important nuances

- Primary key over-fitting example
- Decision trees and prism rules can become too big; Naive Bayes may not be independent

Exploratory Analysis

- Mean, Median, Mode, Standard Deviation, Percentiles, Quintiles, Minimum, Maximum

Look for heavy-tailed or bimodal distributions

- **Continuous**: Histograms
- **Discrete**: Buckets for the most and least frequent values

Appendix: Naïve Bayes for Continuous Outcomes

<u>Example</u>

Temperature	-->	Play?

$$P(play \mid T) = \frac{P(play)\ P(\boldsymbol{T} \mid play)}{P(play) * P(\boldsymbol{T} \mid play) + P(no\ play) P(\boldsymbol{T} \mid no\ play)}$$

Model $P(\boldsymbol{T} \mid play)$ and $P(\boldsymbol{T} \mid no\ play)$ by using the dataset.

*>> One solution is to treat **T** as normally distributed within each case (play and no play).*