Sicurezza Informatica

Fraud Analytics

Instructor: Rossano Schifanella

@SUISS

Why is important?

- A typical organization **loses 5 percent of its revenues** to fraud each year (www.acfe.com).
- The total cost of insurance fraud in the US **> $40 billion** per year (www.fbi.gov).
- Fraud is costing the United Kingdom **£73 billion** a year (National Fraud Authority).
- Credit card companies “lose **~seven cents per every hundred dollars** of transactions due to fraud” (Andrew Schrage, Money Crashers Personal Finance, 2012).

Definition of Fraud

Fraud is an **uncommon**, **well-considered**, **imperceptibly concealed**, **time-evolving** and often **carefully organized** crime which appears in many types of forms.

List of Fraud Types

- Credit Card fraud
- Insurance fraud
- Corruption
- Counterfeit
- Health care fraud
- Telecommunications fraud
- Money laundry
- Click fraud
- Identity Theft
- Tax Evasion
- Plagiarism

Fraud Triangle

Fraud Triangle

- **Pressure**
  - An individual will commit fraud because a pressure or a problem is experienced of financial, social, or any other nature, and it cannot be resolved or relieved in an authorized manner.
- **Opportunity**
  - The precondition for an individual to be able to commit fraud.
- **Rationalization**
  - The psychological mechanism that explains why fraudsters do not refrain from committing fraud and think of their conduct as acceptable.

Example of Rule System

IF:

Amount of claim is above threshold **OR**

Severe accident, but no police report **OR**

Severe injury, but no doctor report **OR**

Claimant has multiple versions of the accident **OR**

Multiple receipts submitted

THEN:

Flag claim as suspicious **AND**

Alert fraud investigation officer.

Disadvantages of Rule based systems

- **Expensive to build**
  - require advanced manual input
  - difficult to maintain and manage
- **Have to be kept up to date**
- Every signaled case requires **human follow-up and investigation**
- Keeping the rule base **lean** and **effective**
  - deciding when and which rules to add, remove, update, or merge
- Fraudsters can **learn the rules** and devise inventive workarounds
- **<u>New emerging fraud patterns are not automatically signaled</u>**

Phases

- **Fraud Detection**
- **Fraud Prevention**

Types of Data

- **Transactional data**
  - structured and detailed information capturing the key characteristics of a customer transaction
- **Contractual**, **subscription**, or **account data**
- **Sociodemographic**
- **Surveys**
- **Behavioral information**
  - preferences of customers, usage information, frequencies of events, trend variables, and so on.
- **Unstructured data**: text, comments, reviews
- **Contextual** or **network** information

Data-driven Fraud Detection: Properties

- **Precision**
  - Statistically based fraud-detection methodologies offer an increased detection power compared to classic approaches.
- **Operational efficiency**
- **Cost efficiency**
- **Interpretability**
- **Regulatory compliance**

Descriptive Analytics

Finding behavior that deviates from normal behavior, or in other words at **detecting anomalies**.

Date (m/d)	Time	Day	Duration	Origin	Destination
1/01	10:05:01	Mon	13 mins	Brooklyn, NY	Stamford, CT
1/05	14:53:27	Fri	5 mins	Brooklyn, NY	Greenwich, CT
1/08	09:42:01	Mon	3 mins	Bronx, NY	White Plains, NY
1/08	15:01:24	Mon	9 mins	Brooklyn, NY	Brooklyn, NY
1/09	15:06:09	Tue	5 mins	Manhattan, NY	Stamford, CT
1/09	16:28:50	Tue	53 sec	Brooklyn, NY	Brooklyn, NY
1/10	01:45:36	Wed	35 sec	Boston, MA	Chelsea, MA
1/10	01:46:29	Wed	34 sec	Boston, MA	Yonkers, MA
1/10	01:50:54	Wed	39 sec	Boston, MA	Chelsea, MA
1/10	11:23:28	Wed	24 sec	White Plains, NY	Congers, NY
1/11	22:00:28	Thu	37 sec	Boston, MA	East Boston, MA
1/11	22:04:01	Thu	37 sec	Boston, MA	East Boston, MA

Outlier

"An outlying observation, or **outlier**, is one that appears to **deviate markedly** from other members of the sample in which it occurs."

Anomalies Detection

- Allow detecting fraud that is **different in nature from historical fraud**
- Descriptive techniques however show to be **prone to deception**, exactly by the camouflage-like fraud strategies
- Requires extensive **follow-up** and **validation** of the identified, suspicious observations (<u>not all outliers are frauds</u>)

Unsupervised Methods

Unsupervised Methods

Example: Credit Card

Example Credit Card Transaction Data Fields

Transaction ID	Transaction type	Date of transaction
Time of transaction	Amount	Currency
Local currency amount	Merchant ID	Merchant category
Card issuer ID	ATM ID	Cheque account prefix

Outliers Detection

possible fraud

possible fraud

Example: Credit Card

Outliers Detection

possible fraud

possible fraud

Unsupervised Methods

- **Statistical Outlier Detection**
  - z-scores
  - Grubbs test
  - Fit a distribution, or mixture of distributions and label the observations with small values for the probability density function as outliers.
- **Break-Point Analysis**
  - sudden change in account behavior, which merits further inspection

Unsupervised Methods

- **Peer-Group Analysis**
  - A peer group is a **group of accounts that behave similarly to the target account**
  - The peer group of a particular account needs to be identified
  - The behavior of the target account is contrasted with its peers using a **statistical test** such a Student's t-test, or a **distance metric** such as the Mahalanobis distance

Unsupervised Methods

Peer-Group Analysis

A key advantage of peer-group vs breakpoint analysis is that it tracks anomalies by considering inter-account instead of intra-account behavior.

Unsupervised Methods

Peer-Group Analysis

For example, if one was to compare transaction amounts for a particular account with previous amounts on that same account (intra-account), then the spending behavior during Christmas will definitely be flagged as anomalous. By considering peers instead (inter-account), this problem is avoided

Unsupervised Methods

- **Clustering**
  - split up a set of observations into segments such that the homogeneity within a segment is maximized (cohesive), and the heterogeneity between segments is maximized (separated)
    - Clustering transactions in a credit card setting
    - Clustering claims in an insurance setting
    - Clustering tax statements in a tax-inspection setting
    - Clustering cash transfers in an anti-money laundering setting

What is Cluster Analysis?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Examples

- Group together documents, for example similar news on a website
- Group genes together that have a similar function
- Group similar trajectories
- Group research project together
- Group countries together

To keep in mind

Clustering

Definition of distance/ similarity + **Method**

↓

Many (important) choices. Euclidean distance for now

Euclidean Distance

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2}$$

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Manhattan Distance

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Partitional Clustering

Original Points

A Partitional Clustering

Hierarchical Clustering

Hierarchical Clustering I

Dendrogram

Hierarchical Clustering II

Dendrogram

Clustering Algorithms

- **K-means and its variants**
- **Hierarchical clustering**
- **Density-based clustering** – not discussed

Partition with predefined number of clusters

- Given a database of **n** objects, a partitioning method constructs **n** partitions of the data (**k ≤ n**).
  - Each partition represents a **cluster**.
- Example: **k-means**

K-means

Start with **K centroids** (how do we choose them? Will see)

K-means

Assign points to each centroid based on **proximity**

K-means

Recompute the centroids

Illustration (animated)

K-means

Reassign the points based on the new centroids, etc, until you reach convergence (do we reach it?)

K-means

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

Predictive Analytics

- **Learn from historical observations** in order to retrieve patterns that allow differentiating between normal and fraudulent behavior.
- These techniques exactly **aim at finding silent alarms**, the parts of their tracks that fraudsters cannot cover up.
- Supervised learners can be applied to **predict** or **detect** fraud as well as to **estimate the amount** of fraud.

Predictive Analytics

- Disadvantages:
  - **They need historical examples to learn from** (i.e., a labeled data set of historically observed fraud behavior)
  - **Reduced detection** power with respect to **drastically different fraud types** making use of new mechanisms

Supervised Methods

- **Regression**
  - predicting amount of fraud, e.g, [0,1] or [0, ∞]
- **Classification**
  - Binary (fraud vs no-fraud)
  - Multiclass (sever fraud, severe fraud, medium fraud, no fraud)
- In fraud detection, **both classification and regression models can be used simultaneously**.
  - Expected fraud loss (EFL) = PF × LGF

PF = probability of fraud (classification)
LGF = Loss given fraud (regression)

Regression

A regressor is a mapping

$$\hat{f} : \mathscr{X} \rightarrow \mathbb{R}$$

The regression learning problem is to learn a function estimator from examples **(x_i, $f(x_i)$)**.

Classification

A classifier is a mapping:

$$\hat{c} : \mathscr{X} \to \mathscr{C}$$

where

$$\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$$

the “hat” over the name of the classifier denotes that the classifier is an approximation of the true but unknown function **c**.

An example is a pair:

$$(x, c(x)) \in \mathscr{X} \times \mathscr{C}$$

where **x** is an “instance” and **c(x)** is the true class of the instance (possibly contaminated by noise).

Classification

Learning a classifier involves constructing the function such that it matches **c** as closely as possible.

Scoring classifier

A scoring classifier is a mapping:

$$\hat{\mathbf{s}} : \mathscr{X} \to \mathbb{R}^k$$

the boldface notation denotes that the output is a vector, i.e.:

$$\hat{\mathbf{s}}(x) = (\hat{s}_1(x), \ldots, \hat{s}_k(x))$$

where the i-th component is the score assigned to class $\mathbf{c_i}$ for instance **x**. If we only have two classes, it usually suffices to consider the score for only one of the classes.

Here the scores are a measure of the **confidence** the classifier has in its prediction.

Class probability estimation

A class probability estimator is a scoring classifier that outputs probability vectors over classes, i.e., a mapping:

$$\hat{\mathbf{p}} : \mathscr{X} \rightarrow [0,1]^k$$

We write:

$$\hat{\mathbf{p}}(x) = \left(\hat{p}_1(x), \ldots, \hat{p}_k(x)\right)$$

where the i-th component is the probability assigned to class $\mathbf{C_i}$ and

$$\sum_{i=1}^{k} \hat{p}_i(x) = 1$$

If we have only two classes, ther $\hat{p}(x)$ denotes the estimated probability for the positive class.

Target definition

- **Hard to determine (not absolute certainty)**
  - insurance fraud: small amount claims are most likely not further considered, even if they are fraudulent

tax fraud: Key Company is the perpetrator of the fraud. It starts up a side company (Side Company 1), which makes revenue but deliberately does not pay its taxes and hence intentionally goes bankrupt. And so on...

hard to distinguish a regular bankruptcy due to insolvency from a fraudulent bankruptcy due to malicious intent

all fraudulent companies go bankrupt, but not all bankrupt companies are fraudulent
not error-proof

Supervised Methods

- **Linear Regression**

- **<u>K-Nearest Neighborhood</u>**
- **<u>Decision Trees</u>**
  - variable selection as variables that occur at the top of the tree are more predictive of the target
  - high-level segmentation
  - white-box model with a clear explanation behind how it reaches its classifications.
- **Neural Network**
- **Support Vector Machines**

Nearest Neighbor Classifiers

K-Nearest-Neighbor Strategy

- Given **object x**:
  - **Find the k most similar objects to x (k-nearest neighbors)**
    - **Variety of distance or similarity measures** can be used
    - This requires comparison between x and all objects in the database (**expensive!**)

K-Nearest-Neighbor Strategy

- **Classification**
  - Find the class label for each of the k-neighbor
  - Use a **voting** or **weighted voting** approach to determine the **majority class** among the neighbors (**a combination function**)
    - **Distance-based**
      - closer neighbors get higher weights
    - **Heuristic**
      - weight for each neighbor is based on domain-specific characteristics of that neighbor
  - Assign the majority class label to x

K-Nearest-Neighbor Strategy

- **Prediction**:
  - Identify the **value of the target attribute** for the k-neighbors
  - Return the **weighted average** as the predicted value of the target attribute for x

Geometric Interpretation

Boundaries created by a 1-NN classifier.

Model Complexity

k-NN Properties

- **Memory-based**, no explicit training or model, **"lazy learning"**
  - defers all of the work until new instance is obtained
- Less data preprocessing and model evaluation, but **more work has to be done at classification time**
- In its basic form, one of the most **simple** machine learning methods
- Can be used as a **baseline method**.

k-NN Algorithm Advantages

- **Easy to understand and program**
- **Explicit reject option**
  - if there is no majority agreement
- **Easy handling of missing values**
  - restrict distance calculation to subspace
- Extremely **flexible** classification scheme
- Well suited for
  - **Multi-classes**
  - Records with **multiple or ambiguous class labels**
- Can sometimes be the best method!

k-NN Algorithm Disadvantages

- **Affected by local structure**
- **Sensitive to noise, irrelevant features**
- **Computationally expensive**
- **Large memory requirements**
- More frequent classes dominate result (if distance not weighed in)
- **Curse of dimensionality**:
  - high number of dimensions and low number of training samples
    - "nearest" neighbor might be very far
    - in high dimensions "nearest" becomes meaningless

Model Evaluation

- **Hold-Out Validation**
- **Cross Validation**
  - k-cross fold validation
  - leave-one-out cross-validation.

Hold-out

1
Data
Labels
Training Data
Training Labels
Test Data
Test Labels
2
Training Data
Training Labels
Hyperparameter Values
Learning Algorithm
Model

Splitting in Training and Testing Data: Stratification

- Simple process of **random subsampling**.
  - Our dataset is a random sample drawn from a probability distribution; and we typically assume that this sample is **representative of the true population**
  - **Further subsampling without replacement** alters the statistic (mean, proportion, and variance) of the sample.
- **Stratification** randomly splits the dataset so that **each class is correctly represented in the resulting subsets**
- Not a big concern if we are working with large and balanced datasets
- Rule of thumb: in many occasions helps with k-fold cross validation (Kohavi 1995)

Repeated Hold-out Validation

- Called also **Monte Carlo Cross-Validation**
- Repeat the holdout method **k times** with different random seeds

$$\mathrm{ACC}_{avg} = \frac{1}{k}\sum_{j=1}^{k} ACC_j,$$

This work by Sebastian Raschka is licensed under a
Creative Commons Attribution 4.0 International License.

Hyperparameters and Model Selection

This work by Sebastian Raschka is licensed under a
Creative Commons Attribution 4.0 International License.

Example of K-Nearest Neighbours

(probably not the best since K-NN is a lazy learner and a nonparametric method)

Three-Way Holdout Method for Hyperparameter Tuning

Three-Way Holdout Method for Hyperparameter Tuning

3

4

Three-Way Holdout Method for Hyperparameter Tuning

This work by Sebastian Raschka is licensed under a
Creative Commons Attribution 4.0 International License.

K-Fold Cross Validation

This work by Sebastian Raschka is licensed under a Creative Commons Attribution 4.0 International License.

All data for training and testing

Test folds in k-fold cross-validation are **not overlapping**

In repeated holdout:

- performance estimates becomes dependent; it can be problematic for statistical comparisons

- some samples may never be part of the test set (k-fold cross validation guarantees that each sample is used for validation)

Online Evaluation

- **A/B testing** (more **info**)
- **Multiarmed bandits** (more **info**)

Metrics For Classification

- **Accuracy** (default in scikit-learn)
- **Per-Class Accuracy**
- **Confusion Matrix**
- **Precision**, **Recall**, **F1-score** and **Support**
  - (**classification_report** function in scikit-learn)
- **ROC Curves** and **AUC**

Confusion Matrix

		ACTUAL	
		POSITIVE	NEGATIVE
PREDICTED	POSITIVE	True Positive (TP)	False Positive (FP) Type I Error
	NEGATIVE	False Negative (FN) Type II Error	True Negative (TN)

Metrics For Classification

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

specificity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

precision

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Metrics For Classification

accuracy (ACC) = 1 - Error Rate

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

F1 score is the harmonic mean of precision and sensitivity

$$F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Metrics For Classification

- The **support** is the number of occurrences of each class in the true data.
- The **per-class accuracy** computes the **accuracy** for each class independently
  - different from the average accuracy amongst all the classes
  - **useful in case of unbalanced classes** or when **the classifier performed very well/bad only for specific classes**

Metrics for Regression

- **R^2** or **coefficient of determination** (default in scikit-learn)
  - indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s)

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

total sum of squares

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

residual sum of squares

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

- **Root Mean Squared Error**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Decision Trees

- How can we **select one or more attributes/ features/variables that will best divide the samples with respect to our target variable** of interest?
  - Middle-aged professionals who reside in New York City on average have a churn rate of 5%
    - What is the segment?
    - What is the target variable?

Supervised Segmentation

Attributes:

head-shape: square, circular

body-shape: rectangular, oval

body-color: gray, white

Target:

write-off: yes/no

Which of the attributes would be the best to segment these people in groups such that write-offs will be distinguished from non-write-offs?

Resulting groups should be as pure as possible!

Group Purity

Assume this was our entire data

We can obtain **pure** groups splitting by **body-type**

A measure of purity: entropy

$$\textbf{entropy} = -p_1 \log(p_1) - p_2 \log(p_2) - \ldots$$

where $\mathbf{p_i}$ is the relative percentage of property **i** within the set

p(non-write-off) = 7 / 10 = 0.7
p(write-off) = 3 / 10 = 0.3
entropy(S) $= -0.7 \times \log_2(0.7) - 0.3 \times \log_2(0.3)$
$\approx -0.7 \times -0.51 - 0.3 \times -1.74$
$\approx$ **0.88**

Entropy measures the **general disorder** of the set, ranging from

- $\mathbf{p_i = 0}$ at **minimum disorder** (the set has members all with the same property) to
- $\mathbf{p_i = 1}$ at **maximal disorder** (the properties are equally mixed)

A measure of purity: entropy

Information gain

- We would like to measure **how informative an attribute is with respect to our target**
  - how much gain in information it gives us about the value of the target variable.
- **Measures the change in entropy due to any amount of new information being added**
- **attribute k values:**
  - original set of examples the **parent set**
  - result of splitting the **k children sets**

Information gain

$$\mathbf{IG(parent, children) = entropy(parent) - [p(c_1)*entropy(c_1)+p(c_2)+entropy(c_2)+ \ldots]}$$

where $\mathbf{c_i}$ is a child derived from the splitting

entropy(parent) ≈ 0.99

entropy(Balance < 50K) ≈ 0.39

entropy(Balance ≥ 50K) ≈ 0.79

IG = entropy(parent)-

[p(Balance<50K)*entropy(Balance<50K) +

p(Balance≥50)* entropy(Balance ≥ 50K)]

≈ 0.99 - [0.43 × 0.39 + 0.57 × 0.79]

≈ 0.37

the split reduces entropy substantially

entropy(parent) ≈ 0.99

entropy(Residence=OWN) ≈ 0.54

entropy(Residence=RENT) ≈ 0.97

entropy(Residence=OTHER) ≈ 0.98

IG ≈ 0.13

this split (Residence) reduces entropy less than the previous case (Balance)

How did we come up with this?

Discretization

- Equal interval
- Equal frequency
- K-means

Binary Split that maximizes gain

More info at this link

(Data Mining, Prof. Dr. J. Fürnkranz)

Example: Attribute Selection

- The Audubon Society Field Guide to North American Mushrooms
- 5,644 samples (2,156 poisonous, 3,488 edible mushrooms)
- 23 features

Attribute name	Possible values
CAP-SHAPE	bell, conical, convex, flat, knobbed, sunken
CAP-SURFACE	fibrous, grooves, scaly, smooth
CAP-COLOR	brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
BRUISES?	yes, no
ODOR	almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
GILL-ATTACHMENT	attached, descending, free, notched

Attribute name	Possible values
GILL-SPACING	`close, crowded, distant`
GILL-SIZE	`broad, narrow`
GILL-COLOR	`black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow`
STALK-SHAPE	`enlarging, tapering`
STALK-ROOT	`bulbous, club, cup, equal, rhizomorphs, rooted, missing`
STALK-SURFACE-ABOVE-RING	`fibrous, scaly, silky, smooth`
STALK-SURFACE-BELOW-RING	`fibrous, scaly, silky, smooth`
STALK-COLOR-ABOVE-RING	`brown, buff, cinnamon, gray, orange, pink, red, white, yellow`
STALK-COLOR-BELOW-RING	`brown, buff, cinnamon, gray, orange, pink, red, white, yellow`
VEIL-TYPE	`partial, universal`
VEIL-COLOR	`brown, orange, white, yellow`
RING-NUMBER	`none, one, two`
RING-TYPE	`cobwebby, evanescent, flaring, large, none, pendant, sheath ing, zone`
SPORE-PRINT-COLOR	`black, brown, buff, chocolate, green, orange, purple, white, yellow`
POPULATION	`abundant, clustered, numerous, scattered, several, solitary`
HABITAT	`grasses, leaves, meadows, paths, urban, waste, woods`
EDIBLE? *(Target variable)*	`yes, no`

You have to advice Christopher McCandless whether to eat a mushroom or not, before he starves.

He has time only for a brief call on the phone, and you can ask about only of the characteristics of the mushroom: **gill-color**, **spore-print-color**, **odor**.

Which one you choose?

Figure 3-6. Entropy chart for the entire Mushroom dataset. The entropy for the entire dataset is 0.96, so 96% of the area is shaded.

Figure 3-7. Entropy chart for the Mushroom dataset as split by GILL-COLOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

Figure 3-8. Entropy chart for the Mushroom dataset as split by SPORE-PRINT-COLOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

Figure 3-9. Entropy chart for the Mushroom dataset as split by ODOR. The amount of shading corresponds to the total (weighted sum) entropy, with each bar corresponding to the entropy of one of the attribute's values, and the width of the bar corresponding to the prevalence of that value in the data.

Decision Tree

- A flow-chart-like **tree structure**
- **Internal node denotes a test on an attribute**
- **Branch represents an outcome of the test**
- **Leaf nodes represent class labels or class distribution**

- Decision tree generation consists of **two phases**
  - **Tree construction**
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - **Tree pruning**
    - Identify and remove branches that reflect noise or outliers

Decision Tree

- **Classifying an unknown sample**
  - Test the attribute values of the sample against the decision tree

Tree Induction

Tree Induction

Tree Induction

<u>Very rare</u> situation in which all the leaves are pure!

Tree Induction

- **Exponentially** many decision trees can be constructed from a given set of attributes
- Finding the most accurate tree is **NP-hard**
- In practice: **greedy algorithms**
  - Grow a decision tree by making a series of **locally optimum decisions** on which attributes to use for partitioning the data

Tree induction as a set of rules

IF (Body shape=Rectangular) AND (Body Color=White) THEN **Class=YES**

IF (Body shape=Rectangular) AND (Body Color=Black) THEN **Class=NO**

IF (Body shape=Oval) AND (Head Shape=Square) THEN **Class=NO**

IF (Body shape=Ovale) AND (Head Shape=Circle) THEN **Class=YES**

Probability Estimation

- Assign to each leaft **an estimate of the probability of membership** in the different classes
- Tree induction can easily produce probability estimation trees instead of simple classification trees
- **Frequency-based** estimate of class membership: if a leaf contains **n** positive and **m** negative instances, the probability of any new instance being positive may be estimated as **n/(n+m)**.

+ write-off
o No write-off

Probability Estimation: Issues

- Attention to the probability of class membership for segments **with very small numbers of instances**.

- At the extreme, if a leaf happens to have only a single instance, should we be willing to say that there is a 100% probability that members of that segment will have the class that this one instance happens to have

Advantages of Decision Trees

- **Inexpensive** to construct
- Requires **no prior assumptions**
- **Extremely fast at classifying** unknown records
- **Easy to interpret** for small-sized trees
- Tree is **easy to visualize**
- **Accuracy is comparable** to other classification techniques for many **simple** data sets

Disadvantages of Decision Trees

- **Many simplifications** introduced to make computation feasible (stopping conditions, pruning)
- Can be very **non-robust**.
  - A small change in the training data can result in a big change in the tree, and thus a big change in final predictions.
- High risk of **overfitting**
- Subject to statistical errors

Social Network Analytics

- Learn and detect characteristics of fraudulent behavior in a **network of linked entities**
- Include an extra source of information in the analysis, being the **relationships between entities**, and as such may contribute in uncovering particular patterns indicating fraud.

Fraud Cycle

- **Fraud detection**: Applying detection models on new, unseen observations and assigning a fraud risk to every observation.
- **Fraud investigation**: A human expert is often required to investigate suspicious, flagged cases given the involved subtlety and complexity.
- **Fraud confirmation**: Determining true fraud label, possibly involving field research.
- **Fraud prevention**: Preventing fraud to be committed in the future.

Fraud Cycle

Automated detection algorithm
Fraud detection
Fraud investigation
Fraud confirmation
Fraud prevention
Current process

Frequency of retraining

- The **volatility** of the fraud behavior
- The **detection power of the current model**
- The **amount of (similar) confirmed cases** already available in the database
- The **rate** at which new cases are being confirmed
- The required **effort to retrain the model**

Example: Credit Card

- Having information on which transaction was fraudulent or not, we can apply a predictive model
- Additional behavioral patterns would be discovered, e.g.:
  - Small purchase followed by a big one
  - Large number of online purchases in a short period
  - Spending as much as possible as quickly as possible
  - Spending smaller amounts, spread across time

References

- Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques, Wiley & SAS Business Series

Questions?

@rschifan

schifane@di.unito.it

http://www.di.unito.it/~schifane