Analysing Insurance Data or The Advantage of TF/IDF Features

Katharina Morik, Hanna Köpcke
Univ. Dortmund, Computer Science Department, LS VIII
{morik,koepcke}@ls8.informatik.uni-dortmund.de
http://www-ai.cs.uni-dortmund.de

Abstract

The input representation is one of the crucial factors of learnability. For knowledge discovery in databases, the transformation of the given data into an appropriate representation is the task of preprocessing and there most efforts are spent. It is our goal to support the design of the overall chain of steps in preprocessing. In this paper, we present a practical application together with its design process. This example case might inspire case designers in a similar situation to try the same procedure that we detected in the end. The key to successful predicting surrender of insurance policies from insurance data was the transformation into a frequency representation. This transformation delivered a condensed data set which we characterise as a TCat-concept [Joachims, 2002]. The learnability of TCat-concepts by a support vector machine can be shown before running it. However, Joachim's theory can only be applied after the transformation. In order to support designers of knowledge discovery cases, we propose a heuristic which can be evaluated on the untransformed data in the database to decide whether the transformation should be applied.

1 Introduction

The success of knowledge discovery depends on several aspects, among them the quality and completeness of the data, the clarity of the task, and the availability of efficient tools. Whereas research on efficient tools for the data mining step has a long tradition, the preprocessing steps have come into focus only recently. Preprocessing inspects data characteristics (e.g., class distribution), enhances the quality of data (e.g., handling of missing values or biased sampling), selects an algorithm for the data mining step, and transforms the given data into the format which the algorithm requires. Some preprocessing tasks such as, e.g., feature generation and selection have raised some attention. Similar to the investigation of a data mining algorithm, research on preprocessing algorithms deal with a single step of the overall knowledge discovery process. However, the steps are interdependent: we choose an algorithm which best fits the data characteristics, but we also change the data such that a given algorithm becomes applicable. For instance, we use a decision tree learner in order to replace missing values by the predicted ones before we apply it or a support vector machine for data mining [Morik and Scholz, 2003]. Focussing on this design task may provide us with insight into principles that relate data characteristics, their transformation, and properties of algorithms. At least, cases of successful knowledge discovery can be used as a guideline for the design of new, similar cases.

In this paper, we want to present a case whose solution was hard to find [1]. We were provided with data from an insurance company in anonymous form. The task was to predict surrender in terms of a customer buying back his life insurance. The classification into possible surrender or continuation would be used in order to select those customers for further actions who are likely to re-buy their contract, and in order to calculate the financial deposits needed in order to pay the re-bought contracts. Data about customers, about contracts and about the components of contracts were given in a time-stamped manner. The generation and selection of relevant features from the given data was the primary task. The choice of the appropriate algorithm for the data mining step is dominated by the chosen feature set. We experimented with decision tree learning, association rule learning, and the support vector machine (MYSVM) [Rüping, 2000]. We present the first trials which did not exceed a precision of 57%. A precision of 96.8% was only achieved when using new features for the changes in contracts that were generated in analogy to term frequency and inverse document frequency (TF/IDF).

We explain the astonishingly good result by theoretical findings about the learnability of text classification by support vector machines [Joachims, 2002]. Now, that we have transformed the data into TF/IDF feature vectors, we can see how they fit Joachim's TCat model. A posteriori we can base our discovery results on sound principles. For the apriori support of the design process we propose a heuristic that is calculated on the given time-stamped data.

The paper is organized as follows. First, we describe the application in Section 2. Second, in Section 3 we describe the first experiments focusing on handling the history of contracts. Third, we describe the successful case in Section 4.1 and in Section 4.2 analyse it in terms of the TCat model and the distribution of the transformed input space. An estimate of the data space after transformation is derived (Section 4.3). We conclude by a proposal to gather successful cases of knowledge discovery at an abstract level and discuss related work in Section 5.

[1] Internal studies at the insurance company found that for some attributes the likelihood of surrender differed significantly from the overall likelihood. However, these shifts of probabilities cannot be used for classification.

2 The Insurance Application

The database consists of 12 tables with 15 relations between them. The tables contain information about 217586 contracts, 533175 components and 163745 customers. The contracts belong to five kinds of insurances: life insurance, pension insurance, health insurance, incapacitation insurance and funds bounded insurance. Every contract consists on average of 2 components. Overall the table of contracts has 23 columns and 1 469 978 rows whereas the table of components has 31 columns and 2 194 825 rows. Each contract and each component may be changed throughout the period of an insurance. For every change of a contract or a component there is an entry in the database with the new values of the features, a unique mutation number, a key representing the reason of change and the date of change. Each contract is on average changed 6 times, each component on average 4 times.

Figure 1 shows an extract of the table storing the contracts to give an idea how the data looks like. The table has the following attributs (not all are shown in the figure):

- VVID is the key identifying a contract.
- VVAENDNR is the unique mutation number
- VVWIVON gives the begin of the validity period of the current record
- VVWIBIS gives the end of the validity period of the current record
- VVAENDART is the key representing the reason of change (for example 1 = rejected application, 4 = cancellation of the mutation expiry, 22 = expiry of premium payment, 110 = inclusion of a person)
- VVAENDDAT gives the date of change
- VVVERSART is the kind of insurance (life, pension, health, incapacitation or funds bounded)
- VVWAE is the currency in which benefits and premiums are paid
- VVSTACD gives the status of the contract
- VVPRFIN states how the premium is paid
- VVPRZA is a technical field
- VVINKZWEI states the number of premium payments per year
- VVBEG gives the begin date of the insurance contract
- VVEND gives the end date of the insurance contract
- VVINKPRL gives the amount of the premium
- VVINKPRE gives the amount of a single allocation
- VVABVB identifies the insurance agent
- VVABGA identifies the insurance agency
- VVSTIFCD is the provision foundation
- VVVORSCD is the kind of provision
- VVBVGCD states whether the insured has a claim for a company pension plan
- VVEUCD states the amount of benefits in case of disability
- PDID identifies the product

Surrender is only observed in 7.7 % of the contracts. Hence, the first characteristic of the case is that it is an example of so-called skewed data [Bi *et al.*, 2001]. Overall 2 181 401 attributes are present in the data warehouse which makes the data very high dimensional. Second, the data are sparse in the following sense: if the attribute values would be transformed into binary attributes, many of the attribute values would be zero. Third, those attribute values occurring frequently do so in the surrender class as well as in the regular class. These characteristics, of course, remind us of the text classification characteristics established by Joachims [Joachims, 2002] and as we will see later on, the analogy actually holds. We found out that attributes concerning the customer did not support the classification into surrender and not surrender. Relevant are the attributes describing the contract and its components. The fourth characteristic is that the data show the history of a contract. Changes of a contract and/or its components are time-stamped. This means, that the same contract is described in several rows of the contract table, each row describing one state of the contract with an attached date.

3 First Experiments

There are many ways to handle time-related data, e.g., [Blockeel *et al.*, 2001; Das *et al.*, 1998; Mannila *et al.*, 1995; Morik, 2000].

It is hard to select the appropriate approach. The first choice that is often successful, is to ignore the time information (Section 3.1). In our case, this means to select for each contract the latest valid record. The second choice is modelling explicitly the sequential structures (Section 3.2). In our case, this means that for each attribute of a contract, the begin date and the end date of a particular attribute value is written as a time interval. The third choice is to compile the time information into the representation. Here, we counted for each attribute how often its value changed within one contract (Section 4).

3.1 Predicting surrender without time information

Feature selection from the given database attributes is hardly obtained without reasoning about the application domain. Our first hypothesis was, that data about the customers could indicate the probability of surrender. Therefore, we applied decision tree learning and MYSVM to customer data, sampling equally many customers who continued their contract and those who re-bought their contract. The data were preprocessed in two ways: the age of a customer was grouped into 20 years classes and a binary attribute *surrender* was generated from the raw data. The resulting set of eleven attributes was transformed into the input formats of the learning algorithms. Decision tree learning achieved a precision of 57% and a recall of 80%. MYSVM obtained a precision of 11% and a recall of 57% with its best parameter setting (radial kernel) [Bauschulte *et al.*, 2002]. Trying association rule learning with the conclusion fixed to surrender, did deliver correlated attributes. However, these were the same correlations that could be found for all customers [Bauschulte *et al.*, 2002]. The description of customers in the database does not entail the relevant information for predicting surrender.

Changes in a customer's situation, e.g., buying a house, marriage, or child birth is not stored. These events can only indirectly be observed by changes of the contract or its components. Hence, we focused in a second experiment on the modification tables related with contract components. 33 database attributes for each component of a contract were selected. Attributes of a component were history of a contract

history of another contract

VVID	VVAENDNR	VVWIVON	VVWIBIS	VVAENDDAT	VVAENDART	...
16423	1	1946	1998	1946	1000	
16423	2	1998	1998	1998	27	
16423	3	1998	1998	1998	4	
16423	4	1998	1998	1998	54	
16423	5	1998	1998	1998	4	
16423	6	1998	9999	1998	61	
5016	1	1997	1999	1997	33	
5016	2	1999	2001	1999	33	
5016	3	2001	2001	2001	33	
5016	4	2001	2001	2001	33	
5016	5	2001	2002	2001	81	
5016	6	2002	9999	2002	94	
...						

Figure 1: Extract of the contract table

combined with a year (1960 - 2002) stating when this attribute was changed. Some of the 1386 combinations did not occur. The resulting table of 991 columns shows in each row the complete set of changes of a component and whether the corresponding contract was re-bought, or not. The tables were transformed into the input format of the learning algorithms. Learning association rules with the conclusion fixed to surrender clearly showed the peak of changes at 1998 where the Swiss law changed and many customers re-bought their contracts. Other changes were just the procedure of surrender, such as finishing the component and the payment. Using MYSVM , 44% precision and 87% recall were achieved using a linear kernel. These results show either that the data do not entail relevant information for surrender prediction, or the representation prepared for learning was not well chosen. They were sufficient, however, to select from the overall data set those attributes that describe contracts and no longer search within the customer data for reasons of surrender.

3.2 Predicting surrender on the basis of time intervals

Taking into account the time aspect of the contract changes was considered an opportunity to overcome the rather disappointing results from previous experiments. There, time was just part of an attribute name. Now, we represented time explicitly. The time stamps of changes were used to create time intervals during which a particular version of the contract was valid. Relations between the time intervals were formulated using Allen's temporal relations [Allen, 1984]. Following the approach of Höppner [Höppner, 2002] who applies the APRIORI algorithm to one windowed time series [Agrawal *et al.*, 1993] a modification to sets of time series has been implemented [Fisseler, 2003]. For each kind of an insurance, association rules about time intervals and their relations were learned according to two versions of time. The first version handles the actual dates, finding that according to a change of Swiss law many customers bought back their contracts around 1998. The second version normalises the time according to the start of the contract such that time intervals of changes refer to the contract's duration. A restriction of the learning set filtering out the contracts around 1998 intended to prevent the analysis from misleading effects of the law change. The prediction of surrender was tried on the basis of both, component data and a combination of component and contract data, applying biased sampling such that surrender and continuation became equally distributed. The rules learned from the set of histories leading to surrender and the rules learned from the set of continued contract/component histories did not differ. The same interval relations were valid for both sets. Hence, the features representing a sequence of changes did not deliver a characterisation of surrender. We are again left with the question whether there is nothing within the data that could be discovered, or whether we have just badly represented the data for our task.

4 Using TF/IDF Features

In the first experiments we have taken into account the customer attributes, the contract attributes, the component attributes and the time intervals for the states of components and/or contract attributes. However, each change of a component was handled as one event. Frequencies were only counted for all contract histories during the data mining step in order to calculate the support. Counting the frequencies of changes within a contract could offer the relevant information. It would be plausible to conclude that very frequent changes of a contract are an effect of the customer not being satisfied with the contract. If we transform the chosen excerpt from the raw data (about contracts) into a frequency representation, we possibly condense the data space in an appropriate way. However, we must exclude the frequencies of those changes that are common to all contracts, e.g. because of a change of law. The feature from statistical text representation formulates exactly this: term frequency and inverse document frequency (TFIDF) [Salton and Buckley, 1988]. Term frequency here describes how often a particular attribute a_i of c_j, the contract or one of its components, has been changed.

$$tf(a_i, c_j) = \| \{x \in time\,points \mid a_i\, of\, c_j\, changed\} \|$$

The document frequency here corresponds to the number of contracts in which a_i has been changed. The set of all contracts is written C.

$$df(a_i) = \| \{c_j \in C \mid a_i \, of \, c_j \, changed\} \|$$

Hence the adaptation of the TFIDF text feature to contract data becomes for each contract c_j:

$$tfidf(a_i) = tf(a_i, c_j) log \frac{\| C \|}{df(a_i)}$$

This representation allows one to classify contracts into those that are re-bought and those that are continued. For data in TF/IDF a support vector machine is a promising algorithm to choose.

4.1 Preprocessing

14 attributes from the original database were selected. We transformed the values of the attribut which indicates the reason for a change of a contract (VVAENDART) into binary values. For each of the 121 different values a new attribut with the name MUTx, where x is the attribut value, was constructed. Thus we obtained 134 features describing changes of a contract. To calculate the TFIDF values for these features we considered the history of each contract. We ordered all mutations of a contract by its timestamps and compared two successive mutations with each other. For the 13 original attributes we obtained the term frequency by counting how often the value of the features changed. Figure 2 illustrates the procedure. For the 121 newly created features we counted how often they occurred within the mutations. Figure 3 shows how the calculation was done. It was now easy to calculate the document frequencies for each feature as the number of contracts with a term frequency greater than 0.

4.2 Results

Since surrender is only observed in 7.7% of the contracts we needed a strategy to handle this bias. First we experimented with biased samples. MYSVM learned on random samples with 1000 and 2000 examples and tested on the remaining instances of the population. The best result we achieved with this strategy was a precision of 96.8%, an accuracy of 98.6%, and a recall of 86% using a linear kernel. Our second approach was to perform a 10-fold crossvalidation on 10000 examples punishing the negative examples by a factor of 3. This led to a precision of 94.9%, an accuracy of 99.4%, and a recall of 98.2% on the test set.

4.3 Explanation

The original data space was first reduced by selecting only the contract data. However, this excerpt still is not well suited for learning. The transformation into a frequency representation allows to model the data as TCat-concepts. TCat-concepts model text classification tasks such that their learnability can be proven [Joachims, 2002].

Definition of TCat-concepts: *"The TCat-concept*

$$TCat([p_1 : n_1 : f_1], ..., [p_s : n_s : f_s])$$

describes a binary classification task with s sets of disjoint features. The i-th set includes f_i features. Each positive example contains p_i occurrences of features from the respective set, and each negative example contains n_i occurrences. The same feature can occur multiple times in one document." [Joachims, 2002]

In order to describe the newly constructed data set in terms of TCat-concepts, we need to partition the feature space into disjoint sets of positive indicators, negative indicators and irrelevant features. Using the simple strategy of selecting features by their odds ratio, there are 2 high-frequency features that indicate positive contracts (surrender) and 3 high-frequency features indicating negative contracts (no surrender). Similarly, there are 3 (4) medium-frequency features that indicate positive contracts. In the low-frequency spectrum there are 19 positive indicators and 64 negative indicators. All other features are assumed to carry no information.

To abstract from the details of particular contracts, it is useful to define what a typical contract for this task looks like. An average contract has 8 features. For positive examples on average 25% of the 8 features come from the set of 2 high-frequency positive indicators while none of these features appear in an average negative contract. The relative occurence frequencies for the other features are given in Table 1. Applying these percentages to the average number of features, this table can be directly translated into the following TCat-concept.

```
TCat ( [2 : 0 : 2], [1 : 4 : 3],        # high frequency
       [3 : 1 : 3], [0 : 1 : 4],        # medium frequency
       [1 : 0 : 19], [0 : 1 : 64],      # low frequency
       [1 : 1 : 39]                     # rest
     )
```

The learnability theorem of TCat-concepts [Joachims, 2002] bounds the expected generalization error of an unbiased support vector machine after training on n examples by

$$\frac{R^2}{n+1} \frac{a + 2b + c}{ac - b^2} \quad (1)$$

where R^2 is the maximum Euclidian length of any feature vector in the training data, and a, b, c are calculated from the TCat-concept description as follows:

$$a = \sum_{i=1}^{s} \frac{p_i^2}{f_i} \quad b = \sum_{i=1}^{s} \frac{p_i n_i}{f_i} \quad c = \sum_{i=1}^{s} \frac{n_i^2}{f_i}$$

$a = 5.41, b = 2.326, c = 5.952$ can be calculated directly from the data. The Euclidian length of the vectors remains to be determined. We want to see whether the data transformation condenses the data properly. For texts, Zipf's law gives the approximation [G.K.Zipf, 1949]. Experimental data suggests that Mandelbrot distributions [Mandelbrot, 1959]

$$tf_i = \frac{c}{(k + r)^{\phi}}$$

with parameters c, k and ϕ provide a better fit. For the contract data figure 5 plots term frequency versus frequency rank. The line is an approximation with $k = -0.6687435$ and $\phi = 1.8$. We see that (as is true for text data) also the contract data can be shrinked by the frequency transformation.

$$R^2 = \sum_{r=1}^{d} \left(\frac{c}{(r + k)^2} \right)^2$$

We bound $R^2 \leq 37$ according to the Mandelbrot distribution and come up with the bound of the expected error according to equation 1 of $\frac{22}{n+1}$, consequently after training on 1000 examples the model predicts an expected generalization error of less than 2.2%. It turns out that the transformed data sets can easily be separated by a support vector

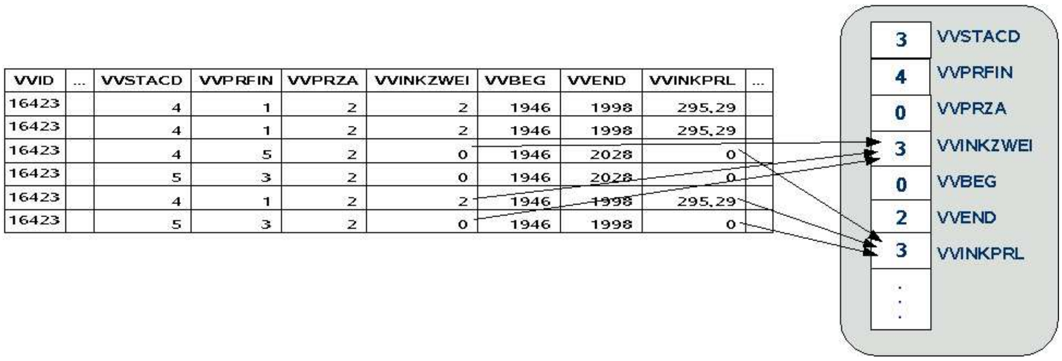

Figure 2: Calculating the term frequency for the original attributes

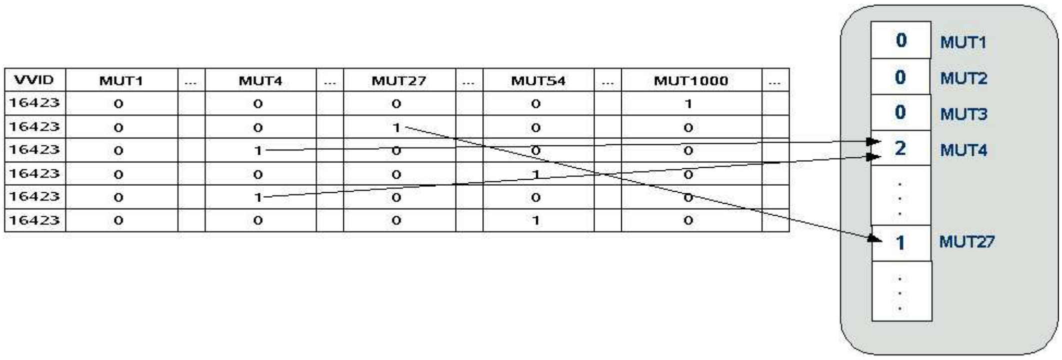

Figure 3: Calculating the term frequency for the newly created features

machine. Hence, the good learning results (0.6% error) are explained.

However, we cannot apply the TCat model to the given data. In order to ease the design process of knowledge discovery we need some hint when the TF/IDF representation is worth the transformation effort. In other words, we should know before the transformation whether the data space will be condensed, or not. We order the original table with time stamps such that the states of the same contract are in succeeding rows. We consider each contract c a vector and calculate the frequency of changes for each of its n attributes $a_1...a_n$ in parallel "on the fly". We can then determine in one database scan the maximum value of the Euclidian length of a vector:

$$\hat{R} = max\left(\sqrt{\sum_{i=1}^{n} tf(a_i, c_j)^2}\right) \quad (2)$$

If $\hat{R} \leq \sqrt{n}m$ where m is the maximum frequency, the transformation into TF/IDF features is worth a try. In our case $n = 14$ and $m = 15$ so that $\hat{R} = 22,913$ which is in fact less than $\sqrt{14}15 = 56.12$.

5 Related Work and Conclusion

In this paper, we have presented a knowledge discovery case together with its design process. The design process can be viewed as the search for a data space which is small enough for learning but does not remove the crucial information. The first approaches and background knowledge could effectively focus the search for an excerpt of the given database on contract data. Feature selection could not be guided in the usual wrapper manner [Kohavi and John, 1997], because no subset of the original features allows for successful learning. The reason lies in the handling of time-stamped data. Only using the current state of a contract abstracts away the crucial information. Following the approach of [Mannila *et al.*, 1995] by explicitly representing time intervals during which a contract had a certain state was not effective, either. The key idea to solving the discovery task was to create TF/IDF features for changes of contracts. We explained the good learning results when using these features by Joachim's learnability of TCat-concepts. In [Domeniconi *et al.*, 2002] the solution of a event prediction is also explained by Joachim's theory. They argue on the basis of the data characteristics similar to our description in Section 2. We move beyond this by actually computing the TCat-concept of our application. In addition, their argument justifies to select a support vector machine for learning. Our focus is on the preprocessing, particularly on the feature construction. In our view, the original idea in the event prediction case was the feature generation for m different event types observed in event sequences: moving a window over the sequence, an event-by-window matrix of size $m \times n$ is created where n is the number of monitor windows. If the number of occurences of an event type within a window is taken as the value of the new feature, then also in the event prediction task there is a frequency encoding. The feature construction of the event prediction case differs from our feature construction. In particular, it does not take into account an analogy to the document frequency as we do. The substantial difference between the paper of Domeniconi and colleagues and ours lies in the focus of the argument. They focus on the usefulness of a support vector machine for their learning task. We focus on how to transform a data set such that learning

	high frequency		medium frequency		low frequency		
	2 pos.	3 neg.	3 pos.	4 neg.	19 pos.	64 neg.	39 rest
pos. contract	25%	12.5%	37.5%	0%	12.5%	0%	12.5%
neg. contract	0%	50%	12.5%	12.5%	0%	12.5%	12.5%

Table 1: Composition of an average positive and an average negative contract

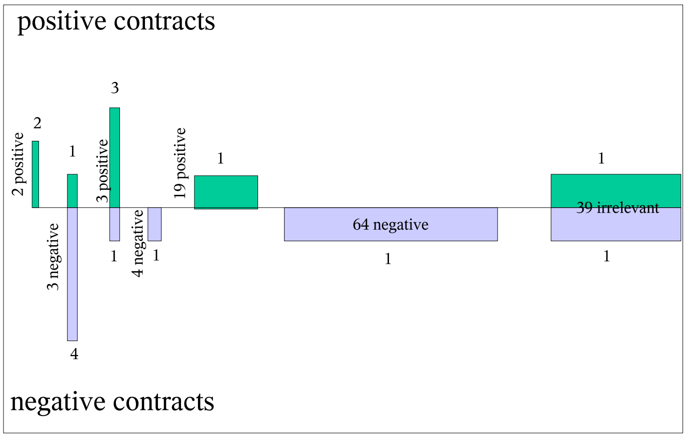

Figure 4: TCat-Concept for the contract data

becomes possible. We derive from the characteristics of the transformed data which learning algorithm to choose.

We are tackling the subject of how to guide the design of knowledge discovery processes. Our subject is the overall sequence of steps. In the application described here, the steps are:

- selecting contracts as the relevant aspect for surrender prediction,
- selecting 14 attributes from the contract tables,
- counting the frequencies of the features in each contract (term frequency),
- counting the frequencies of features in all contracts (document frequency),
- creating a table with the target attribute “surrender” and the 134 features of TF/IDF type,
- splitting the set of records for cross validation,
- running MYSVM on the training set, and
- evaluating the learning result on the test set.

There are some approaches taking into account the overall process. The IDEA system plans operator sequences on the basis of applicability and outcome conditions and evaluates alternative algorithms [Bernstein *et al.*, 2002]. This approach applies to small datasets and the tools of weka [Witten and Frank, 2000]. In our applications to existing very large databases, the sequences become too large and there are too many choices for a planning approach. For similar reasons, the multi-agent approach that develops a chain of steps is not applicable in our case [Zhong *et al.*, 2001].

We cannot provide case designers with automatic planning in the space of all possible features and algorithms. However, we can offer case designers

- successful cases of knowledge discovery documented at the meta-level,
- means for the adaptation of an abstract case to a new data set,
- operators for the preprocessing steps,
- applicability conditions for operators that can be tested on the meta-data or the original database and
- heuristic tests for the usefulness of certain feature generation operators.

The MiningMart system already provides users with all services except for the last one [Morik and Scholz, 2003][2]. In this paper, we investigated the last point, i.e. how to develop heuristics for the usefulness of frequency features. As always, a good theory is the best practice. The success of this knowledge discovery process can be explained using Joachim’s proof of the learnability of TCat-concepts. This theory, however, does not tell us, when to transform given data into the form of TF/IDF features – where the theory applies. We proposed an efficient estimate for when you might want to transform the given attributes into TF/IDF features and calculate the TCat-concept in order to estimate the learnability.

Acknowledgment We thank Jörg-Uwe Kietz and Regina Zücker for their information about the insurance practice and the anonymised database. For stimulating discussions on the support vector machine, mathematics, and representation languages we thank Stefan Rüping wholeheartedly.

References

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington, D. C., may 1993.

[Allen, 1984] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[2] For more details, the list of involved partners, and the system itself you might visit *www.mmart.cs.uni-dortmund.de*. Submitting a successful case described at the meta-level is very much appreciated!

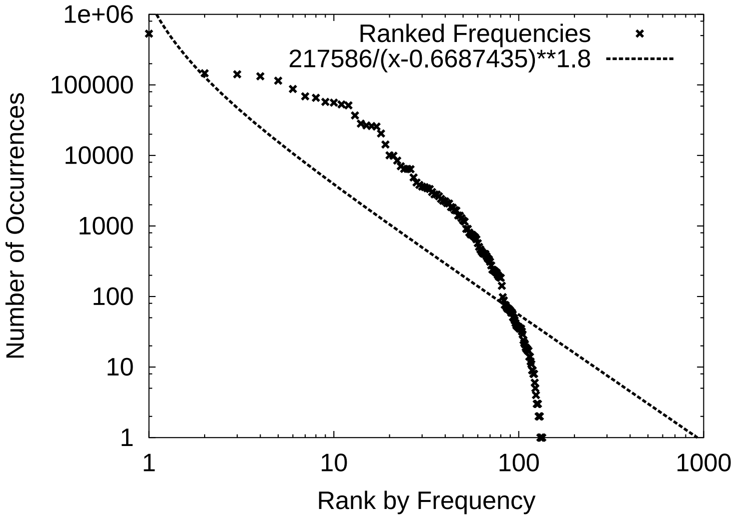

Figure 5: Distribution of term frequencies in the contract data on a log log scale. The line is an approximation of the observed curve using a Mandelbrot distribution.

[Bauschulte *et al.*, 2002] Fabian Bauschulte, Ingrid Beckmann, Stefan Haustein, Christian Hueppe, Zoulfa El Jerroudi, Hanna Koepcke, Phillip Look, Katharina Morik, Boris Shulimovich, Klaus Unterstein, and Daniel Wiese. PG-402 Endbericht Wissensmanagement. Technical report, Fachbereich Informatik, Universität Dortmund, 2002.

[Bernstein *et al.*, 2002] Abraham Bernstein, Shawndra Hill, and Foster Provost. An intelligent assistant for the knowledge discovery process. Technical Report IS02-02, New York University, Leonard Stern School of Business, 2002.

[Bi *et al.*, 2001] Zhiqiang Bi, Christos Faloutsos, and Flip Korn. The DGX distribution for mining massive, skewed data. In *7th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2001.

[Blockeel *et al.*, 2001] Hendrick Blockeel, Johannes Fürnkranz, Alexia Prskawetz, and Francesco Billari. Detecting temporal change in event sequences: An application to demographic data. In Luc De Raedt and Arno Siebes, editors, *Proceedings of the 5th European Conference on the Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Computer Science*, pages 29–41. Springer, 2001.

[Das *et al.*, 1998] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule Discovery from Time Series. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 16 – 22, New York City, 1998. AAAI Press.

[Domeniconi *et al.*, 2002] Carlotta Domeniconi, Chang shing Perng, Ricardo Vilalta, and Sheng Ma. A classification approach for prediction of target events in temporal sequences. In Tapio Elomaa, Heikki Mannoila, and Hannu Toivonen, editors, *Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Artificial Intelligence. Springer, 2002.

[Fisseler, 2003] Jens Fisseler. Anwendung eines Data Mining-Verfahrens auf Versicherungsdaten. Master's thesis, Fachbereich Informatik, Universität Dortmund, Januar 2003.

[G.K.Zipf, 1949] G.K.Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, 1949.

[Höppner, 2002] Frank Höppner. Discovery of Core Episodes from Sequences. In David J. Hand, Niall M. Adams, and Richard J. Bolton, editors, *Pattern Detection and Discovery*, volume 2447 of *Lecture notes in computer science*, pages 1–12, London, UK, September 2002. ESF Exploratory Workshop, Springer.

[Joachims, 2002] Thorsten Joachims. *Learning to Classify Text using Support Vector Machines*, volume 668 of *Kluwer International Series in Engineering and Computer Science*. Kluwer, 2002.

[Kohavi and John, 1997] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence, Special Issue on Relevance*, 1–2:273–324, 1997.

[Mandelbrot, 1959] Benoit Mandelbrot. A note on a class of skew distribution functions: Analysis and critique of a paper by H.A.Simon. *Informationi and Control*, 2:90 – 99, 1959.

[Mannila *et al.*, 1995] H. Mannila, H. Toivonen, and A. Verkamo. Discovering frequent episode in sequences. In *Procs. of the 1st Int. Conf. on Knowledge Discovery in Databases and Data Mining*. AAAI Press, 1995.

[Morik and Scholz, 2003] Katharina Morik and Martin Scholz. The MiningMart Approach to Knowledge Discovery in Databases. In Ning Zhong and Jiming Liu, editors, *Handbook of Intelligent IT*. IOS Press, 2003. to appear.

[Morik, 2000] Katharina Morik. The representation race - preprocessing for handling time phenomena. In Ramon López de Mántaras and Enric Plaza, editors, *Proceedings of the European Conference on Machine Learning 2000 (ECML 2000)*, volume 1810 of *Lecture*

Notes in Artificial Intelligence, Berlin, Heidelberg, New York, 2000. Springer Verlag Berlin.

[Rüping, 2000] Stefan Rüping. *mySVM-Manual*. Universität Dortmund, Lehrstuhl Informatik VIII, 2000. http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/.

[Salton and Buckley, 1988] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[Witten and Frank, 2000] Ian Witten and Eibe Frank. *Data Mining // Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, 2000.

[Zhong *et al.*, 2001] N. Zhong, C. Liu, and S. Ohsuga. Dynamically Organizing KDD Processes. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3):451–473, 2001.