ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume 63 244 Number 6, 2015
http://dx.doi.org/10.11118/actaun201563062229

CUSTOMERS' OPINION MINING FROM EXTENSIVE AMOUNT OF TEXTUAL REVIEWS IN RELATION TO INDUCED KNOWLEDGE GROWTH

Jan Žižka[1], Arnošt Svoboda[2]

[1] Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic
[2] Department of Applied Mathematics and Computer Science, Faculty of Economics and Administration, Masaryk University, Žerotínovo nám. 617/9, 601 77 Brno, Czech Republic

Abstract

ŽIŽKA JAN, SVOBODA ARNOŠT. 2015. Customers' Opinion Mining from Extensive Amount of Textual Reviews in Relation to Induced Knowledge Growth. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis,* 63(6): 2229–2237.

Customers of various services are often invited to type a summarizing review via an Internet portal. Such reviews, written in natural languages, are typically unstructured, giving also a numeric evaluation within the scale "good" and "bad." The more reviews, the better feedback can be acquired for improving the service. However, after accumulating massive data, the non-linearly growing processing complexity may exceed the computational abilities to analyze the text contents. Decision tree inducers like *c5* can reveal understandable knowledge from data but they need the data as a whole. This article describes an application of windowing, which is a technique for generating dataset subsamples that provide enough information for an inducer to train a classifier and get results similar to those achieved by training a model from the entire dataset. The windowing results, significantly reducing the complexity of the learning problem, are demonstrated using hundreds of thousands reviews written in English by hotel-service customers. A user obtains knowledge represented by significant words. The results show classification accuracy errors, training and testing time, tree sizes, and words relevant for the review meaning in dependence on the training subsample size. Finally, a method of suitable training-set size estimation is suggested.

Keywords: text mining, customer opinion analysis, decision trees, decision rules, windowing, large data volumes, machine learning, computational complexity, training-set size

INTRODUCTION

One of the present-day typical tasks is the analysis of web-based data, apparently whatever scientific or application area is taken into consideration. Collecting reviews from customers of various services is gaining in popularity because it can serve as a valuable feedback from the comparative advantage acquiring point of view, Žižka and Rukavitsyn (2012). Intuitively, the more customers' meaningful opinions are available, the better information and knowledge – coming from the feedback – may be applied to the service improvement, thus supporting the competitive capabilities of economic subjects, Dařena *et al.* (2014).

Unfortunately, there is a couple of reasons why such an idea contends with difficulties when it should be realized in the real world. Customers are human beings that use their so called *natural language* for their communication, including sharing or providing opinions, be it reading and listening (passively) or talking and writing (actively). In the following, this research report deals with the customers' opinions that have a common, unstructured form expressed as freely written, not too long textual documents. Providers of various services make today possible to use personal

computers and the Internet to let the customers write their opinions. Such entries are stored, can be again read via the Internet by other (maybe potential future) customers, and analyzed for revealing relevant points, which play significant roles in the opinions from different viewpoints. In this paper, the semantic analysis – performed by computers – was the main goal of the presented research.

Putting the existence of hundreds of natural languages aside, two other weighty problems must be taken into consideration when thinking about mining information and knowledge from the data, which has the non-numerical textual form expressed in natural languages (words, phrases, sentences):

- machines do not really understand human natural languages, and
- data volumes are growing excessively.

The first matter must be sorted out by suitable representation of textual documents written in natural languages. The second issue is related to the problem that – as the other side of the coin – has arisen out of very positive feature of the present day: it is relatively very easy to assemble and store electronic data, which leads to the emerging problem now informally called as *big data*. Thus, the main question here was how computers can discover significant knowledge in large number of customers' textual reviews.

It is, of course, true that in reality much more than 200,000 text documents (mentioned here) would be necessary to analyze; it could be easily millions, billions, and more. At the present time, the overwhelming majority of algorithms and procedures require having all the data in one batch – and the contemporary hardware has big memory difficulties, often unbeatable. That is another topic related to the *big data* problem area, which is out of the scope of this paper. The presented research deals here with a more traditional approach based on so called *windowing* described below.

The following sections provide the description of the textual real-world data obtained from the Internet, which was used for the investigation of the problems briefly mentioned above, including an applied specific data representation, a method of revealing possible information and knowledge via text mining, Srivastava and Sahami (2009), based on machine learning, then the comparison of large experimental results obtained with the help of various parameters, and a brief discussion.

Input Data Description

To investigate the presented problems, this research used data created by customers of hotel services. The data came from a publicly accessible web-site of a popular Internet-based company (booking.com) that enabled customers on-line booking of hotel accommodation almost anywhere. After using the service, customers were allowed to write down their reviews simply from computer keyboards and the reviews were published on the booking company web-site. The customers could use any language and, in essence, no strict limitations were requested. The reviews were written quite freely in tens of different natural languages during several years. Each review had a special mark evaluating the positivity or negativity of the used hotel service: one star for a quite negative opinion, five stars for a quite positive one, plus two to four for mixed experience.

Tab. I illustrates several randomly chosen typical positive and negative reviews. One positive example is in two languages (Polish and English). The last negative example "all was pleasant" probably got just one star by mistake because it looks quite positively (unless it was meant as irony). The shown examples were not corrected, they were here only transformed to the lowercase form, thus words like "luggeges" represent mistyping and grammatical errors (it should be "luggage" and cannot be in the plural). Incorrectly written words artificially increase the dictionary size with a negative impact to the information contents and computational complexity. Sometimes a review could not be categorized without knowing its star labeling, for example that "dining room" sample did not tell much.

English reviews distinctively prevailed, even if not all authors had English as a native language – English was very often taken as a kind of an "international" language. Altogether, there were almost 2,000,000 English positive and negative reviews, where the positive ones were more frequent (almost 1,300,000). Other significantly represented languages included Spanish, French, German, and others according to the native speakers' distribution and number. Sometimes, authors of reviews used their native language together with the same message written in English (mainly smaller nations). The length of reviews was different – the shortest ones had just one word (like "Excellent"), the longest ones up to some 130 words, while the arithmetic average was around 21 words. The reviews were published plainly as they were written by their authors, without any spell-check or grammatical corrections, containing many mistakes that played a role as a kind of signal noise. More details can be found in publications of various text-mining research tasks using the same data (also in more languages), see, for example, Žižka and Dařena (2011).

One of the research goals was to test capabilities of common personal computers (PC's) that could be expected to mine such textual data provided that a regular hotel-service provider would own and apply an ordinary PC. Thus, not all the English reviews were used because the used available computer was a routine office PC with 8 GB of memory (RAM – it was the critical point) and two double-core processors 3 GHz. After some experimenting, a random choice of 200,000

I: *Several illustrative examples of positive and negative reviews*

Positive review examples	Negative review examples
Everything was great in this hotel, the staff, the swimming pool, the size of the room, the kitchen with microwave, coffee maker, toaster, and a hair dryer in the bathroom. Although it is in a very relaxing area, there were bars and supermarkets around the hotel and it was very convenient. There are lots of activities for small kids and excellent size of swimming pool. Definitely recommendable.	Terrible! Very unclean. First. upon arrival we were told that we could not have a non smoking room (which we had booked months ago). If this was not to our liking we could swap the next day! Second. Room was not clean, and of course smelt of smokers. Third. We found a cockroach in the room. Fourth. Internet was charged at $24.95 per 24 hours, but was given to us free after mentioning that my booking through booking.com said it was free… Should i go on?
I like your location not far from old town.	Toilet stink maybe because of the moist tiles & no ventilation. Kettle not provided but tea, coffee etc available.
Clean room and friendly staff, good for costal walk with easy access to local shops and restaraunts.	Tv channels poor, i cant immagine a 4 stars hotel with no one helped us to get our luggeges to the room the reciptionest was very rude by saying u hav to take ur luggege to ur room!!
W oliwkowym gaju nieduża hacjenda z dopracowanymi dodatkami dała nam wyciszenie, nasłonecznienie i pyszne jedzenie. In the olive grove the small hacienda with supplements touched up gave us the calm, the solar exposure and the delicious food.	I am afraid the outer windows of my room (they were double glazed) were extremely dirty and spoiled the view. My other criticism would be that radio 2 was played in the brasserie throughout breakfast. The incessant chattering of chris evans and guests is very annoying when you are trying to read a newspaper or arrange your thoughts for the day. I don't think radio is appropriate in such a setting. Selected background music – if anything – would be more acceptable.
Excellent.	Dining room
Walking distance to the beach and shopping areas. The variety of food is good.	All was pleasant.

(ca 10%) reviews was used for the investigation to avoid the insufficient memory failure. Some of the previous research tasks, for example, Žižka and Dařena (2012), demonstrated that such a choice was quite representative for the given data from the obtained result point of view.

Preprocessing and Representation of Reviews

At the first step, the reviews were divided into two classes: *positive reviews* and *negative reviews*. The positive review class included documents having 5 and 4 stars while the negative one contained reviews with 1 and 2 stars. Reviews with 3 stars were taken as something between and (because their number was not high) were excluded from the processing. To avoid problems with the unbalanced number of positive and negative reviews, the source set of 200,000 reviews contained 50% of positive and 50% of negative ones. Strongly unbalanced classes need additional, not always easy preprocessing, see, for example, Ganganwar (2012).

At the second step of the data preprocessing, the reviews were freed from special characters (for example, *@, #, $, %, *, &*, and so like) if they were there. As various experiments and publications showed, for example, Sebastiani (2002), such characters in this kind of text analysis were semantically quite meaningless, providing no useful information and acting as noise. Similarly, all numbers and punctuation was removed, too, from the same reasons, so only terms (words) having at least one alphabetic character remained. In addition, all letters were converted to the lowercase form. Such kind of preprocessing is common, independently on a language, because it positively (from the computational complexity point of view) decreases the number of lexical units without any significant loss of useful information for a machine. Finally, all words having only one or two letters were removed, too, because such words included many so called *stop-words*, Sebastiani (2002), which did not contribute to the relevant semantic information of the reviews. Those words included mainly articles *a, an*, prepositions like *at, in, on,* and so like. Such simple kind of the data preparation was very beneficial because the total number of unique terms (the dictionary of the reviews) was substantially decreased from almost 120,000 to 56,462. That filtering enabled ultimately carrying out the large experiments with 200,000 reviews using a routine PC.

At the third step, the words in the reviews were transformed into numbers defined as frequencies of particular terms in individual documents. This research used the standard text document representation, Sebastiani (2002): Each review was represented as a vector in a multidimensional vector space defined by all the reviews' words, where each dictionary word constituted an axis (dimension). A vector was a row in a matrix (table), where each column represented a word. A vector's coordinates were defined by a word frequency in a given review. Considering that the total number of words in the dictionary was 56,462 and the average number of a review's words was only ca 21, the vectors were very sparse (most of coordinates were just zeros).

Non-zero coordinates were mostly = 1, not often 2 or more, so the representation suggested the binary one.

It was possible to apply several more-or-less different representations (for example, binary or tf-idf); however, the previous experiments showed that the review analysis gave almost the same results independently on a specific representation. It was eventually the reason why this research used that simple, not very demanding representation, having in mind making the preprocessing of large data easier.

Design of Experiments

The set of experiments was designed to demonstrate what knowledge could be mined from the large data volume provided that it was impossible to analyze the whole set containing 200,000 reviews – which was true for the given PC. One of possible known solutions is working with just part of the whole dataset. That part can be generated by random selection of reviews, forming a subset of an acceptable size both from the memory and computational time point of view. The question is how many elements such a subset should contain – lower number means faster computation without risking the memory insufficiency but less data also means less information.

In this investigation, the goal was to reveal what words were relevant attributes determining the membership of a review in the given two classes. Intuitively, if the total number of words was some 56,462 and the average number of words per review only 21, which is just 0.037%, a low amount of reviews evidently could not provide information enough because many important words would be eliminated. Thus, what is the right number of such reviews? Optimally all reviews should be used but it is not possible because of their too high number. The question is also what computer memory is available and how much time can be devoted to the analysis, which depends on the computational complexity of the analyzing process.

The experiments worked with many subsets created by random selection, starting from 2,000 reviews (1% from 200,000), gradually (and non-linearly) increasing their number up to 160,000 (80%). Each subset was a "window" through which only part of data was visible. This windowing method was taken over from Quinlan (1993). For each analysis, a random selection (always using different random generator seeds) of a certain number of training samples was performed. Then, the same number of randomly selected testing reviews from the remaining cases was chosen. When the number of training samples exceeded 100,000, the rest (from 200,000) was used for testing. The testing samples were used for predicting the expected possible classification accuracy. The accuracy, Acc, is defined by the expression Acc = (TP + TN)/(TP + TN + FP + FN), where TP stands for the numbers of *true positive*, TN for *true negative*, FP for *false positive*, and FN for *false negative* labeling of unlabeled testing (or classified) cases, Witten *et al.* (2011). The accuracy measure served as an empirical proof of the correct generalization of the training samples, that is, the quality of the knowledge acquired from the data.

Decision Trees/Rules Generator as a Text-Mining Tool

The problem of looking for important words was solved using a particular machine learning technique applied to data mining. One of often employed and reliable methods for finding relevant attributes (variables) is supervised learning aimed at training a classifier like a decision tree, Quinlan (1993), Witten *et al.* (2011), because each tree branch contains a series of questions to values of attributes that decide what class (group, category) a review belongs to. If the attributes are words the values of which are expressed as frequencies, then the words selected for generating a decision tree represent those important ones. Note that the same word may be a member of any (or all) analyzed classes, so a combination of words in a tree branch is predominant. The most important word is in the tree root because there the questioning starts and such a word is tested each time (in 100% of classification procedures). On "lower" tree levels, gradually closer to leaves, the decision is more and more detailed; however, such words are not so often tested. In the leaves (at the ends of branches), the final classification decision is given. Only a very small fraction of all words in a given particular dictionary is usually used and such words define the relevant attributes from the classification point of view. In addition, each branch represents also a rule, so it is possible to say which combination of words leads to an appropriate class.

The decision tree is generated via the process of training, where data samples with known classification are used. An individual sample represents only very concrete information on one case but using the inductive process, the generalization of many samples provides the knowledge searched for. Such knowledge is then represented by a tree or set of rules.

For the experiments, the decision tree generator known as *c5*, see Quinlan (2013), was used. The principle is based on an idea that each attribute is tested how well it splits a (heterogeneous) set of samples originating from more classes to (more or completely homogeneous) subsets containing prevailing number of samples (ideally all) belonging to only one class. The *c5* algorithm measures the "purity" of a subset using *entropy* and is directed at minimizing the final entropy the value of which is ideally zero, while the worst case is when a set contains an even number of each class representatives. The subset entropy, $H(X)$, is computed from the number of samples of each class divided by the number of all samples in the subset – that is the a posteriori probability, $p(x_i)$; for more

details of a subset's information gain see, for example, Quinlan (1993):

$$H(X) = -\sum_i p(x_i) \cdot \log_2 p(x_i), \quad (1)$$

where X is a random variable taking real numerical values $x_i \in \mathbb{R}$. Using X, here actually a word taking frequency values, x_i, a set can be divided into two or more subsets. For each word and its generated subsets, the entropy is computed and then the average entropy given by all the subsets. A word generating the lowest average entropy is selected for questioning. The whole process is recursively repeated as long as the entropy decreases. Some words are never selected because they provide no significant entropy decrease. The ultimate selected words contribute most of all to the entropy lowering. For generating decision trees, an alternative measure of purity of a set could be also the *Gini impurity metrics*, *G(X)*, which is a measure of how often a randomly chosen element from a given set would be incorrectly classified if it were randomly labeled according to the distribution of labels in the subset, Breiman *et al*. (1984):

$$G(X) = 1 - \sum_i p^2(x_i) \, . \quad (2)$$

Often, in practice, the results of *H(X)* and *G(X)* – even if it is not a rule – are very similar or identical. Gini impurity is, for example, used in another popular learning algorithm known as *CART* (classification and regression trees); see, for example, Witten *et al*. (2011). The following investigation and its results are based on applying the entropy minimization implemented in the Quinlan's *c4.5/c5* algorithm, which belongs to the most employed and favorite data mining tools incorporated also in various commercial software products.

As for the computational complexity, it is given by an algorithm employed for the requested analysis. For the above mentioned *decision tree* generator, the upper bound of the computational complexity estimation, O[f(*m, n*)], is $m \cdot n^2$, where m is the number of training samples (rows in the matrix) and n is the number of words (the matrix columns) in the dictionary of reviews. With the increasing m, it can also be expected the increasing n (more documents usually provides – to a certain degree – a wider vocabulary). Obviously, the computational time depends quadratically on the word number, so eliminating meaningless words during the data preprocessing phase is important. More detailed information about the time computational complexity related to decision trees may be found in Chikalov (2011). Except the list of relevant words and their significance, the results of experiments gave also the accuracy errors, sizes of trees (a number of tree nodes), and CPU time depending on the number of training samples. The outputs are discussed in the next section.

RESULTS AND DISCUSSION

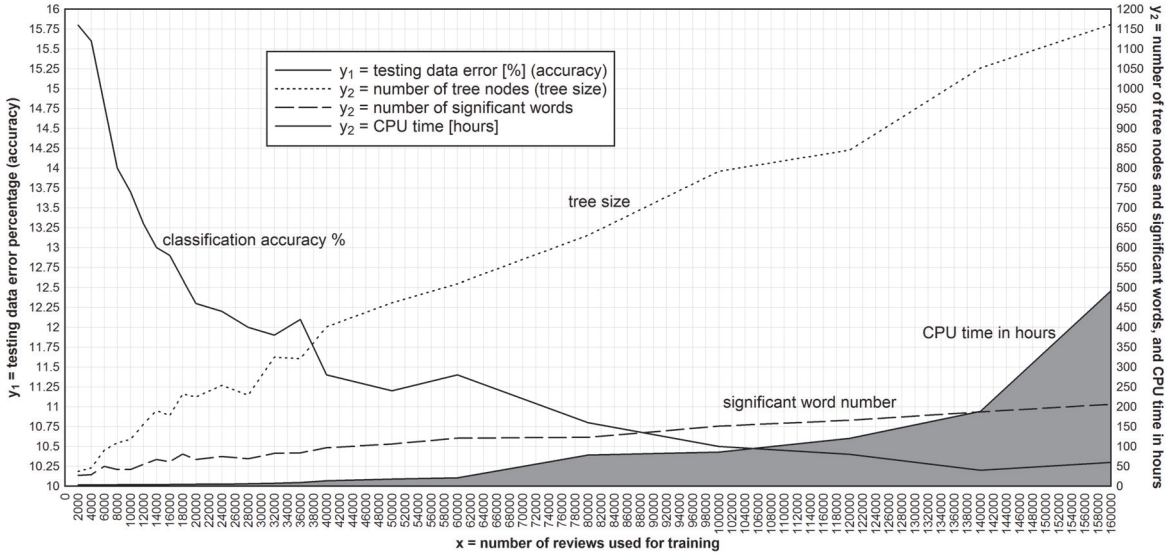
The experiments investigated the influence of the number of training samples on the resulting induced knowledge. As mentioned above, the knowledge was based on the words relevant to the classification/prediction related to the future occurrence of unlabeled customers' textual reviews the number of which increases extremely rapidly. Thus, the main persisting question was: How much the induced knowledge quality (that is, the classification/prediction accuracy) would depend on the volume of the training samples obtained during the past times, taking into account the computational complexity?

Mining the Relevant Features – Significant Words

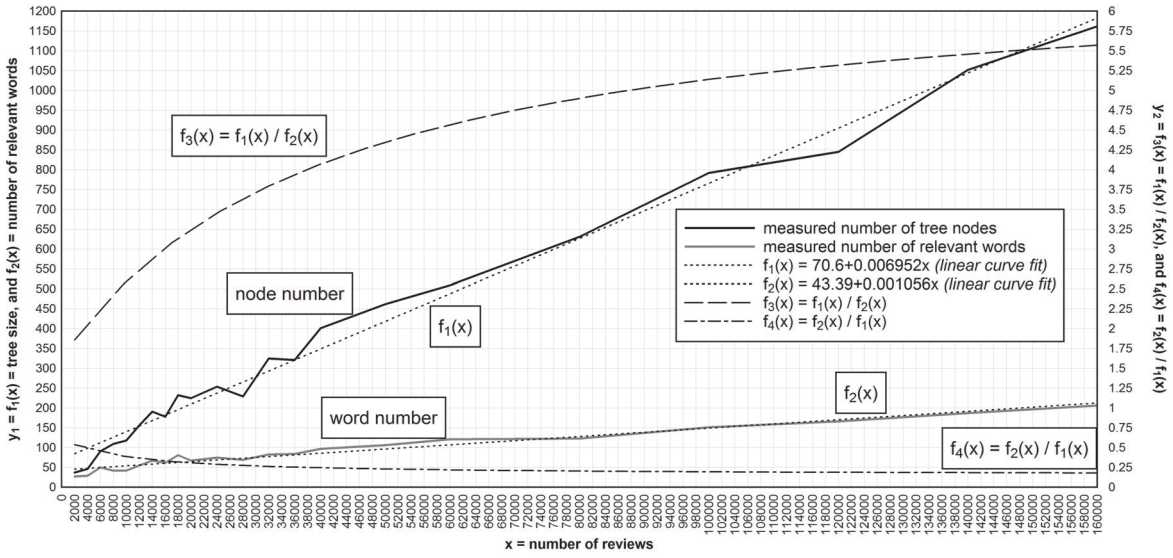
The results of experiments are summarized in graphs Fig. 1 and Fig. 2. The first graph, Fig. 1, shows the influence of the increasing number of training samples (reviews) on the computational time CPU, classification accuracy, as well as on the relevant attribute (significant word) number. The initial smallest number of training reviews was 2,000, and the final largest one was 160,000 from 200,000 (the *x* axis). What is the answer to the question "how many reviews should be used for obtaining good knowledge"? There is apparently no unequivocal answer and it depends on a user's needs as well as possibilities.

As it can be seen, the computational complexity (CPU time) grows very quickly; in the end, the analysis of the 160,000 reviews took almost 500 CPU hours, which is almost three weeks. It is due to the non-linearly increasing computational complexity given mainly by the number of unique words in all the analyzed text documents. The initial review subset with 2,000 samples gave 27 relevant words (*location* used in 100% tree tests – in the root, *excellent* 82%, *friendly* 78%, *located* 72%, *quiet* 71%, *helpful* 69%, *good* 65%, *great* 58%, *close* 56%, *comfortable* 54%, *nice* 52%, *clean* 50%, and so on) after 11,467.6 CPU seconds. The tree size was 37 nodes (after post-pruning) and the accuracy error was 15.8%.

On the other side, the last and largest analyzed review subset had 160,000 samples (80 times larger than the starting subset) and returned 206 relevant words the list of which included the same words as obtained from the lower number of samples plus many others that had their usage just in percentage units (for example, *staff*, *door*, *window*, *windows* 1%, then *well*, *free*, *sleep*, *center*, *room*, *station*, *bathroom* 2%, and so on). The tree size (5.57 times larger) was 1162 nodes (not a simple generalization) and the CPU time 1,771,545.7 seconds (almost 155 times longer). The accuracy error was 10.3% (5.5% decrease). Clearly, the accuracy error decreases fast after adding more and more training samples, however, from a certain data volume the declination is almost negligible in spite of constructing larger and more detailed trees using more words. Somewhere around 55,000



1: *Measured dependence of classification accuracy, tree size, significant word number, and CPU time on the number of reviews*



2: *Linear approximation of the tree-size, f1(x), and word-number, f2(x), dependence on the number of training reviews. The ratio $f_3 = f_1(x)/f_2(x)$ represents the steepness of the mined information growth with respect to the number of training samples*

reviews (CPU 19 hours) the error decreases very slowly, losing just 1% for 160,000 reviews – however, the computational time dramatically increases up to more than 490 hours CPU. Thus, a user has to decide what time is worth of a certain error decrease. Looking at the graph, another question suggests itself: Is it worth of the much extended effort to reach 10% accuracy error instead of 11%? Again, no unequivocal answer can be given as such things depend on a particular application and the data volume.

After the linear approximation of the functions demonstrating the growth of the decision tree and the number of relevant words, the graphs in Fig. 2 show how the steepness of the information gain is gradually decreasing while the tree size (and the word number) constantly grows. The tree grows much faster than the number of relevant words which can be interpreted as the non-linear rate, $f_3(x)$, between these two monitored features. The function $f_4(x)$ is the inverted dependence $f_3(x)$ and represents the relative declination of revealed knowledge (relevant words) with respect to the increasing tree size.

Training Samples' Number and Minimizing the Accuracy Error

Intuitively, the more good training samples are available, the better knowledge may be induced. It is true; however, a very large number of training samples can play also a negative role from the time computational complexity point of view. This

problem was illustrated by Fig. 1 and the non-linear computational time increase is quite evident. While a day or two could be acceptable, weeks and more may be excessive. When 20,000 training samples took just 4.96 CPU hours, 40,000 used 13.68, 80,000 needed 78.3 hours (3.26 days) and 160,000 consumed 492.1 CPU hours, which was 20.5 days. And the accuracy error using the testing data? It was 12.3% for 20,000, 11.4% for 40,000, 10.8% for 80,000, and 10.3% for 160,000 training samples. Naturally, a user should decide what time would be reasonable for her or him. But is it worth to spend almost three weeks (for 160,000 samples) to get the accuracy error (10.3%) lower only by 0.5% than something more than just three days for a half of the training samples volume with the error 10.8%?

In theory, it is possible to find an indirect relationship between the classification error and number of training samples. The *PAC-learning* (probably approximately correct learning), which constitutes the core of the computational learning theory (using the Vapnik-Chervonenkis VC-dimension for continuous attributes), insinuates the answer. As it is out of scope of the research topic described here, no details are given but an interested reader may find a lot of information in, for example, Hastie *et al.* (2009), Hutchinson (1994), and Valiant (1984). Using that PAC theory, it is possible to look for an answer to the question: How to estimate the number of *training* samples, m, which would provide an acceptably low classification error? Thinking on a quite hypothetical classifier, let ε be a given maximum error. Then, the probability that this imperfect classifier would classify correctly m random training samples is given by a simple equation:

$$p(m) \leq (1-\varepsilon)^m. \tag{3}$$

The true is that the probability will decrease very quickly with the increasing number of training samples – and the theory is rather pessimistic. Let h be the number of all possible classifiers in the entire instant space and $k \leq h$ the number of classifiers that make no error on the training samples but are not errorless on the remaining space instances (m training samples is only a subset of all the instances, in practice often very limited). Then, the upper bound $p(m)$ can be easily rewritten in this way:

$$p(m) \leq k(1-\varepsilon)^m \leq h(1-\varepsilon)^m. \tag{4}$$

Because $(1-\varepsilon)^m < e^{-m\varepsilon}$, $p(m) \leq h \cdot e^{-m\varepsilon}$. If the demand would be that $p(m) \leq \delta$, where δ is a certain predefined number, then $h \cdot e^{-m}\varepsilon \leq \delta$, therefore after some adjustment it is possible to write the following:

$$m > \frac{1}{\varepsilon}\left(\ln h + \ln\frac{1}{\delta}\right). \tag{5}$$

The symbol δ represents the probability that a classifier with the error rate $> \varepsilon$ is errorless on the training set. Note that m grows proportionally to $1/\varepsilon$. From the theory point of view, the previous PAC considerations mean that a class is not PAC-learnable in the case when the number m of training samples needed to satisfy the (ε, δ)-requirements is either not realizable (that is, such a large training set is – from any reason – impossible) or the computational time (for a high m) is not acceptable even if such a huge data set would be ready.

Unfortunately, in practice such a procedure will not provide directly the needed m. Anyway, it may be used as a guideline – a kind of the worst-case analysis – when there is a possibility to prepare the training data provided that no insuperable obstacles exist – one can select a classifier, the number of training and testing samples, the time is ample, and so like.

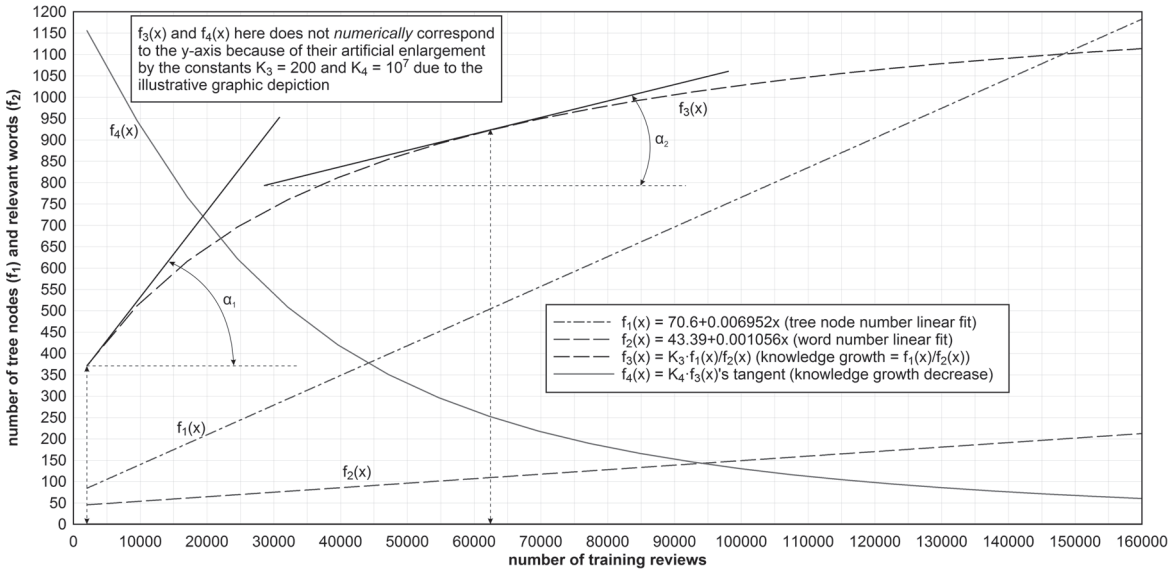
This research experiments investigated the above described text mining problem from the m-and-error point of view. For the given typical textual data, the results are illustrated in Fig. 3. Here, only the linear approximations of the tree size, $f_1(x)$, and relevant word number, $f_2(x)$, were used. The ratio $f_3(x) = f_1(x)/f_2(x)$ illustrates the gradual decrease of the induced knowledge growth with m. Note that the function values for the curves $f_3(x)$ and $f_4(x)$ are intentionally numerically enlarged just to better demonstrate the progress of the training process dependent on m.

Taking a point on the x-axis, a tangent on the corresponding $f_3(x)$ function can be drawn. The tangent angle α gradually decreases, representing the increasing distance between the rapidly growing tree size (number of tree nodes) and the noticeably slowly accumulating number of relevant words that are used for the classification – in Fig. 3, $\alpha_2 < \alpha_1$. Fig. 3 shows the decrease of the knowledge growth, which is also illustrated by the function

$$f_4(x) = \frac{d}{dx} f_3(x);$$

this is de facto the appropriate tangent slope for a given x (the corresponding angle α can be computed using arctangent).

The investigated problem is often linked with data cumulated during longer time periods when the data volume starts to be too big for a standard analysis procedure or the data can change as an analyzed task develops according to certain progressing surroundings, which is typical, for example, also for economy. In such a case, an analyst can make use of a data-window but he/she has to decide what size of the window should be applied. Starting with smaller windows and continuing with larger ones, it is possible to monitor the mined knowledge quality like the classification accuracy and revealed relevant attributes. After reaching an adjusted quality limit, the analyst can save time that grows very rapidly (see Fig. 1) with the increasing window size without providing significant knowledge improvement. One of possibilities is watching the gradual decrease of the induced knowledge growth, $f_3(x)$, related to



3: *The knowledge is here represented by the number of relevant words, $f_2(x)$. This graph clearly demonstrates the gradual decrease of the induced knowledge growth, $f_3(x)$, related to the addition of more and more training samples, which enlarges the tree size, $f_1(x)$. The growth deceleration is illustrated by the declining angle* α *(the tangent angles for certain numbers of training samples*

the addition of more and more training samples, as demonstrated in Fig. 3.

A user should decide when to stop adding new training samples because of their omissible contribution – for example, when after enlarging a number of reviews the information growth $f_3(x_i)$ towards the previous $f(x_{i-1})$ is less than a certain given difference value Δ_{min}: $f_3(x_i) - f_3(x_{i-1}) < \Delta_{min}$ (that is, the f_3 curve starts to be too flat from a certain point x_i on the *x*axis).

CONCLUSION

Not only can the shortage of data be a data mining problem – having too much data may be the cause of difficulty as well. The experimental investigation of the influence of the review number on the knowledge mined from the text documents demonstrated primarily the not surprising cardinal high-time dependence. With the permanent increase of the volume of hotel-service reviews, the CPU time of the text mining process grew strongly non-linearly while the knowledge, expressed in generated semantically relevant words, remained increasing, too, even if its increase was progressively smaller all the time. Among others, the revealed relevant words (or phrases composed of them) can be further used as significant key-words for information retrieval or for defining more detailed topics hidden in text documents.

After finishing the above described research, which aimed at revealing relevant words that represented the reviews, a following series of experiments have been started to mine better knowledge that would provide more information understandable by humans: automatically discovering significant phrases composed from relevant words. To find the phrases, a method of analyzing *n-grams* (here a contiguous sequence of *n* words) was applied to reviews written in English, Spanish, German, and Russian. Similar procedures as described in this article, using the same decision-trees/rules tool, data source, and windows containing constantly 100,000 reviews, were used. From the semantic point of view – unlike 1-*grams* described in this paper – the best phrases were provided by 3-*grams*, for example, "breakfast very good" (a positive phrase), "no free Internet" (a negative phrase) and so like. Details can be found in Žižka and Dařena (2015).

Predictably, the experiments again confirmed the well-known problem: most of today's proven standard machine-learning algorithms need to have all the data in memory at once, which is another obstacle linked with the increasing necessity to analyze and process very big data. Even if a user can employ such sophisticated and well implemented tools as the *c5* decision trees/rules generator, apparently new algorithms that would be able to process very large data (tens and hundreds of millions documents, or much more) not at once but consecutively (in a stream) are necessary, see for example Bidet *et al*. (2010).

REFERENCES

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A. and STONE, C. J. 1984. *Classification and Regression Trees.* Wadsworth International.

BIFET, A., HOLMES, G., KIRKBY, R. and PFAHRINGER, B. 2010. MOA: Massive online analysis. *Journal of Machine Learning Research,* 99(2010): 1601–1604.

CHIKALOV, I. 2011. *Average Time Complexity of Decision Trees*. Springer.

DAŘENA, F., ŽIŽKA, J. and PŘICHYSTAL, J. 2014. Clients' Freely Written Assessment as the Source of Automatically Mined Opinions. *Procedia Economics and Finance,* 12(2014): 103–110.

GANGANWAR, V. 2012. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4): 4247.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd edition. Springer.

HUTCHINSON, A. 1994. *Algorithmic Learning*. Oxford University Press.

QUINLAN, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

QUINLAN, J. R. 2013. Data Mining Tools See5 and C5.0. In: *RuleQuest Research 2013*. [Online]. Available at: https://www.rulequest.com/see5-info.html. [Accessed: March 30, 2015].

SEBASTIANI, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1): 1–47.

SRIVASTAVA, A. N., SAHAMI, M. (eds). 2009. *Text Mining: Classification, Clustering, and Applications*. CRC Press/A Chapman and Hall Book.

VALIANT, L. G. 1984. A Theory of the Learnable. *Communications of the ACM*, 27(11): 1134–1142.

WITTEN, I. H., FRANK, E., HALL, M. A. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd edition. Morgan Kaufmann.

ŽIŽKA, J., DAŘENA, F. 2011. Mining Significant Words from Customer Opinions Written in Different Natural Languages. In: *Lecture Notes in Artificial Intelligence*, 6836(1): 211–218.

ŽIŽKA, J., DAŘENA, F. 2012. Parallel Processing of Very Many Textual Customers' Reviews Freely Written Down in Natural Languages. In: *IMMM 2012: The Second International Conference on Advances in Information Mining and Management.* 147–153.

ŽIŽKA, J., RUKAVITSYN, V. 2012. Automatic Categorization of Reviews and Opinions of Internet E-Shopping Customers. In: *Transdisciplinary Marketing Concepts and Emergent Methods for Virtual Environments*. Hershey, Pennsylvania (USA): IGI Global, 154–163.

ŽIŽKA, J., DAŘENA, F. 2015. Automated Mining of Relevant N-grams in Relation to Predominant Topics of Text Documents. In: *Lecture Notes in Artificial Intelligence*, 9302(1): 461–469. Springer Verlag.

Contact information

Jan Žižka: zizka@mendelu.cz
Arnošt Svoboda: arnost.svoboda@econ.muni.cz