Automatic categorization of reviews and opinions of Internet e-shopping customers

Jan Žižka, Vadim Rukavitsyn

Faculty of Business and Economics, Mendel University in Brno
Department of Informatics/SoNet Research Center
Zemědělská 1, 613 00 Brno, Czech Republic
E-mail: zizka@mendelu.cz, vadichruk@list.ru

Abstract

E-shopping customers, blog authors, reviewers, and other web contributors can express their opinions of a purchased item, problem, film, book, and so like. Typically, there are various opinions centered around one topic (e.g., a commodity, film, etc.). From the Business Intelligence viewpoint, such entries are very valuable, however, difficult to being automatically processed because they are in a natural language. Human beings can distinguish the various opinions. Because of the very large data volumes, could a machine do the same? The suggested method is using the machine-learning (ML) based approach to this classification problem, demonstrating on some real-world data that a machine can learn from examples relatively well. The classification accuracy is better than 70 %, even if not perfect because of typical problems

associated with processing of unstructured textual items in natural languages. The data characteristics and experimental results are shown in enclosed tables and graphs.

Keywords

customer opinion analysis, machine learning, textual data, unbalanced samples, classification

Customer Opinion Analysis in Business Intelligence

Business intelligence (BI) is a specialization connected with technological applications to the automatic support of making decisions and the competition ability in economics. BI deals with the online analytical processing (OLAP), predictions, and data analyses, taking knowledge and information from data. Today, it is typical that there are big volumes of very different data which usually contain very useful, however, hidden information. Revealing this information by traditional analytical and modeling methods is sometimes very difficult not only because of big volumes of data but also, for example, because of non-homogenous nature of this data. Information in databases could be general or special: numerical, nominal, binary, acoustic, image, video, including text in natural languages. This heterogeneous nature introduces certain problems, for example, in creating mathematical models which is often impossible without excessive simplifications. As a good example, a reader can imagine textual comments of clients about some commodities, their purchasing and seling, and so like. Human experts can get information from natural-language data connected with solving a special problem and finding in

the data certain rules or other forms of the previously hidden knowledge: data mining. Modern computer science provides an array of algorithms in the area of artificial inteligence, for example, *machine learning* (see Alpaydin, 2010). One possibility is using a collection of text databases as labeled training samples to teach a machine what and how to do in a specific situation when new unknown but more-or-less similar data items appear in the future: classification or prediction (see Hastie, Tibshirani, and Friedman, 2009). This method is called *supervised learning.* Having knowledge obtained by training a machine can suggest a solution based on a certain similarity to one or more generalized cases known from the past times. Such a machine learning-based approach tries to emulate behavior of human experts (see Rukavitsyn and Žižka, 2010).

Data Description

To investigate possibilities of automatic data-mining from customer comments that are written in a quite free, unstructured form using natural language, the authors collected some publicly accessible textual data from the Internet web-site *amazon.com*. The main intention was to get comments about various consumer goods with at least 100 different opinions per each goods item provided by purchasers. The customer reviews describe their experiences that are good, bad, or something between. It is possible to apply also a certain scale as a kind of classification, or *rating*: from one star (the worst experience) up to five stars (the best one). The reviews are expected to explain reasons of their ratings which are usually relatively short, tens or hundreds of

words. Typically, the language is English, however, with many mistypings, grammar errors, and so like. In addition, the used English is really very "international", and the customers are not only people whose native language is one of existing English languages that can more or less differ in grammar and vocabulary. Also, a reader of reviews can sometimes see non-standard interjections and onomatopoeic words.

The nine different commodities the reviews of which were used in the research are shown in Table 1. Interestingly, the average customer rating is very typically closer to five stars which means that customers were probably mostly satisfied.

Insert Table 1 about here.

For the experiments described further, the data were prepared using a simple, standard approach. For each data-set, all its words created a corresponding dictionary. The dictionary did not include numbers, punctuation, and special characters. All words were transformed into the lowercase format. No linguistic, grammatical, semantic, or prepositional relations and phrases were considered. Such a preprocessing phase is now standard because it removes many difficult linguistic problems stemming from the human-like data as natural languages. On the other side, a textual document loses some information which can be partially compensated by using larger data volumes. People would make bibliographic search certainly better, however, when the data volume is large, it takes too long time and needs trained experts. Thus, every textual review was

transformed into a vector that was sparse because it contained only a small part of the complete dictionary of a particular review area given by a certain commodity. The vector components were numerical frequencies of words in a document (the vectors contained mostly zeroes).

Experiments and Results

Classification

The goal of experiments was to show how a computer can be trained to recognize a class of a review, where the class was a category of a customer opinion. Such an approach takes advantage from a special scientific area called *inductive machine learning*, where a selected algorithm for its training uses samples the category of which is known. After its training, that algorithm can, even if not quite perfectly, recognize the appropriate class of an unlabeled textual document. Such an approach can help a lot for large data volumes provided that the classification accuracy is high. The principal problem is a good and reliable algorithm selection – usually, researchers have to test several algorithms empirically, for specific data-sets, and then choose the best alternative. In addition, different algorithms have various parameters that can influence the results as well. Fortunately, today we can employ several very good software tools for such a research procedure. This investigation used a system named WEKA (see Witten and Frank, 2005) which contains tens of algorithms and is easy to use. The tested algorithms and their results are described in the following tables. For each algorithm, the tables show its accuracy percentage. The accuracy is defined as the ratio between correctly classified testing samples to

all used testing samples. For the testing, the experiments used so called 10-fold cross-validation where the data entries are randomly divided among 10 subsets and training is carried out with 9 subsets while testing with the remaining one. Each subset is successively used for testing and the resulting accuracy is then given by an average value from the ten computations. Thus, this result represents the expected pessimistic error for the future real categorization applications.

The following nine WEKA algorithms (see Witten and Frank, 2005) were tested from their *classification accuracy* point of view: SMO (support vector machines), J48 (an entropy-based decision tree), LADTree (a decision tree with LogitBoost strategy), Random forest and Random tree (decision trees with randomly selected data attributes), IB1 and IBk (*k* nearest neighbors for *k*=1 and *k*=3), Bagging with REPTree (decision trees with information gain), and Naïve Bayes (a probabilty-based algorithm that ignores possible mutual dependencies of individual data attributes). Those algorithms were used with default parameters.

Firstly, the authors tried to classify into five classes, where each class was given the number of stars assigned by customers. The results are summarized in Table 2.

Insert Table 2 about here.

Secondly, the following investigation focused on classification into only two classes: one, two, and three stars were taken as *bad* rating, while four and five stars represented *good* rating.

Table 3 shows the achieved results. Apparently, the results for two classes (strictly satisfied or dissatisfied purchasers) provided higher accuracy in comparison with the previous case that used more precise categorization with five classes. One of the reasons is that each of the five classes had less samples available than in the two-class case, and another reason is that classes overlap to a certain degree and it is not easy to separate the class members so strictly.

Insert Tab 3 about here.

Problems with Unbalanced Training Samples

Hence, the two-class rough-approach results gave lower classification error. However, it is necessary to be careful when evaluating such results. In the case of the *Battery* data-set, the classification now looks significantly improved but the *real* reason is that some 95 % of its data belongs to one class (five and four stars) and 5 % to the second one (one up to three stars). For an algorithm, it is easy to simply assign all samples to one class only and then the error cannot be greater than 5 %. This is a sequel of working with unbalanced training samples (see Kubat and Matwin, 1997; Kubat, Holte, and Matwin, 1997). The remaining data-sets were not affected by that unbalance. It is necessary to take this possibility into account. As a reader can see, the best accuracy was showed by the SMO and Naïve Bayes algorithms. These algorithms generally worked very well also with other tested data-sets.

Unfortunately, the above described straightforward classification approach malfunctions for small and unbalanced data sets, as it was demonstrated for the *Battery* data-set. To solve the mentioned problem, the authors combined all data-sets into a big one. The idea behind was that *positive* and *negative* opinions have to share something common within each category, not so dependent on a specific commodity – customers should use similar words and phrases when describing a bad or good experience with a commodity: *bad or good shoes/bad or good DVD players/bad or good watches*, and so on. This primitive example shows that such words as *bad* or *good* can represent the opinion, even if the reality is, of course, much more complex and difficult for the natural language.

Heretofore, the nine data-sets of rather different commodities were processed. Let us suppose that a data-set, which would be combined from all the data, could be a "universal" training data for every type of a commodity. Also, let us use classification only for two classes, *bad* and *good*. For processing the combined data, only the best algorithms from the preliminary experiments were employed: SMO, Naïve Bayes, Random Forest, Random Tree, and J48. Table 4 displays the results of this experiment. Testing was performed by the same cross-validation mentioned above.

Insert Table 4 about here.

However, the combined data-set had again the notorious problem: It was unbalanced. One class contained 322 training samples, the other one 1904. Unbalanced training data sets represent a known problem in many applications because algorithms tend to favor the strongly predominated class (here, 1904:322, that is, almost 86 % was in one of the two classes), especially from the *accuracy* point of view. Consequently, a user could not rely on such results generated, in principle, by simple statistic, even if a more elaborate method – as *machine learning* – was used because a user would not be always interested in statistical results covering the whole population of possible samples: Individual cases should not be always put in the shade by the dominating majority. Therefore, to avoid similar problems when using real data, the next set of experiments was aimed at creating balanced combined training data-sets.

Using Balanced Combined Training Samples

For testing a real-world problem, the following experiments took a new data-set It was again the publicly available data found on the *amazon.com* web-site. This data-set contained reviews from 15 different types of commodities and had 3650 instances. This data contained 11697 attributes (different words). The words which were mentioned in all instances less than four times were removed as insignificant attributes. Similarly, words like ‘a’, ‘the’, and so like (stop-words), were removed, too. This removing of attributes that contained no classification information resulted in a dictionary having just 4864 words, therefore reducing the computational complexity and ‘noise’ generated by insignificant words. At the same time, only six of roughly balanced training data-sets were selected from the 15 ones. The term *roughly balanced* stand for around

three asterisks on the scale from one to five. Finally, there were 1424 training instances available.

The final experiments used the SMO algorithm with the *Normal PolyKernel* parameter because it provided generally the best results, even if some other algorithms (J48, LADTree, Random Forest, Random Tree, and Naïve Bayes) were tested with good results, too. Using the 10-times 10-fold cross-validation testing, SMO gave almost 80% of the accuracy. Then, the experiments used the same trained algorithms, however, the testing was performed with three different *amazon.com* unbalanced data-sets (see Table 5) that were not employed during the training phase (supplied test sets). The best *accuracy* results are shown in Table 6.

Insert Table 5 about here.

Insert Table 6 about here.

The prediction accuracy for the unbalanced testing data-sets was more than 70 % in every case. It means that using the method described above results into a general picture of customer's opinions about every commodity or service that is covered by the same common dictionary which is built using samples taken from the merged and balanced commodity reviews. It is possible to say that the attitude to a commodity is negative (or positive) with the 70 % accuracy.

If a user would be interested in which words are significant for the categorization of a customer review into one of the predefined classes (here positive and negative), he or she can apply an alternative learning algorithm that provides the mined knowledge in an illustrative form. For example, the knowledge represented by a decision-tree form can show which words split the path to a relevant class according to the queries about specific words. Figure 1 displays the LADTree algorithm trained using the data mentioned in the experiment descriptions. A reader can see such 'typical' words (and their combinations) as *great, not, excellent, poor, money, bad, nothing, easy, negative, back, returned,* that lead to either negative or positive review. A collection of such words for individual classes can also help with various automatic review processing as retrieving negative opinions to analyze possible imperfections, and so like.

Insert Figure 1 about here.

Depending on the number of positive or negative reviews, it is also possible to predict the class membership for a new, unlabeled (non-categorized) review instance. Naturally, more training examples provide generally better accuracy, and vice versa. The following graphs Fig. 2, Fig. 3, and Fig. 4 illustrate the agreement between the prediction and the correct classification. On the horizontal axis named *comments*, there is the number of reviews. On the vertical axis named *points*, there is the number of the positive or negative attitudes to the selected commodity: a positive comment represents +1 point, a negative one represents –1 point. Even if the automatic classification accuracy is not 100 %, a reader can see the general agreement between the real and

predicted review categorization. Often, there is a problem with the correct classification for short text items, and some words can have the different meanings in different situations, based on their context. For example, '*not* bad' and '*not* good' means the different value of the word *not*. Therefore, more training examples can improve the accuracy because it generally covers a larger dictionary containing more information.

Insert Figure 2 about here.

Insert Figure 3 about here.

Insert Figure 4 about here.

Conclusions

In this article, the method of the customer natural language-review processing was described. The processing was based on applying machine-learning tools to classification, where the classes were given by positive and negative review categories. Generally, the training process worked well, reaching mostly the accuracy over 90 %. However, the closer investigation revealed that results were often unreasonably influenced by unbalanced number of training samples for each category. The research described above in this paper solved this problem by creating balanced

sets of training samples. The experiments demonstrated that merging the training samples from customer positive and negative opinions of various commodities could be used for creating a common training set that was also well balanced. The classification accuracy percentage was then lower (above 70 %), however, the classification and prediction was more reliable as the tests with supplied data-sets showed. The main goal was to find a way for the categorization using natural language words expressing the customers' opinions without being influenced by a specific commodity type. The experimental results with certain publicly available real data from *amazon.com* indicated that the selected method could provide good results – the following research is going to investigate additional improvements of the textual data preprocessing to reach better classification results.

References

Alpaydin, E. (2010) *Introduction to machine learning*. MIT Press.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *Statistical Learning: Data Mining, Inference, and Prediction.* Second Edition. Springer Series in Statistics.

Kubat, M., Holte, C. R. and Matwin, S. (1997) 'Learning When Negative Examples Abound'. ECML 1997, pp. 146-153.

Kubat, M. and Matwin, S. (1997) 'Addressing the Curse of Imbalanced Training Sets: One-Sided Selection'. ICML 1997, pp. 179-186.

Rukavitsyn, V. and Žižka, J. (2010) 'Opinion classification in text entries using machine-learning approach'. ISDMCI-2010, pp. 283-288.

Witten, I. H. and Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*. Second Edition. Morgan Kaufmann.

Table 1. The basic features of the nine various *amazon* data-sets.

Data type	Average rating	Review number
Battery Tender Junior 12V Battery Charger	4.5	229
Coffee People, Donut Shop K-Cups for Keurig Brewers	4.5	186
Crocs Cayman Sandal	4.5	192
Eureka 4870MZ Vacuum Cleaner	4	230
Men's Health 1-Year Magazine Subscription	3.5	106
Timex Men's Ironman Triathlon 42 Lap Analog/Digital Dress Watch	4	172
Toshiba 640 GB USB 2.0 Portable External Hard Drive	4.5	156
Twilight: The Complete Illustrated Movie Companion	4.5	298
Wii Nunchuk Controller	4.5	228

Table 2. Percentage of the classification accuracy for five classes.

Data \ Algorithms	Battery	Coffee People	Crocs	Eureka	Men's Health	Timex	Toshiba	Twilight	Wii
SMO	83.52	70.25	69.93	32.83	35.82	42.94	62.89	73.94	70.18
J48	81.27	61.98	63.62	36.08	39.64	39.85	60.20	71.59	67.48
LADTree	82.02	60.77	66.00	42.30	41.36	42.87	57.98	68.02	67.46
Random forest	85.77	68.6	66.50	46.90	39.64	49.74	59.35	77.24	67.45
Random tree	75.66	54.53	60.52	35.32	28.45	39.36	48.12	68.63	60.91
IB1	83.15	69.02	28.76	32.83	32.09	49.71	61.52	76.24	68.25
IB3	85.77	71.10	62.19	27.00	35.73	50.85	61.54	78.23	70.92
Bagging	85.78	71.10	70.86	44.00	51.09	49.67	62.47	78.23	70.52
Naïve Bayes	79.02	64.85	65.14	31.53	39.73	51.43	55.79	67.06	68.98

Table 3. Percentage of the classification accuracy for two classes.

Data / Algorithms	Battery	Coffee People	Crocs	Eureka	Men's Health	Timex	Toshiba	Twilight	Wii
SMO	95.90	87.60	86.93	83.83	67.55	82.87	88.00	94.08	93.03
J48	94.02	83.08	83.07	74.30	69.91	74.23	82.27	94.72	93.43
LADTree	94.76	84.32	85.45	74.77	63.18	82.24	83.16	92.09	89.92
Random forest	95.14	85.95	86.4	78.02	69.64	82.24	80.14	94.4	93.05
Random tree	93.66	78.17	80.19	70.13	67.09	78.46	76.13	90.76	89.17
IB1	93.63	82.22	85.95	69.74	64.09	82.83	81.01	92.76	92.66
IB3	95.14	83.87	86.43	74.30	62.18	82.83	79.68	94.40	93.05
Bagging	95.14	85.97	86.43	80.52	65.82	82.21	86.68	94.4	93.05
Naïve Bayes	95.51	86.82	85.45	77.58	72.64	80.99	84.53	90.76	89.15

Table 4. Accuracy of classification using the selected algorithms and combined data.

Algorithms	Accuracy %
SMO	77.69
J48	77.99
Random forest	83.42
Random tree	76.78
Naïve Bayes	78.26

Table 5. Parameters of unbalanced supplied test data-sets.

Data	Instances	Positive reviews	Negative reviews
1	116	47	69
2	49	13	36
3	107	91	16

Table 6. Classification accuracy for the best algorithms and the data from Table 5.

Data	Algorithm	Accuracy %
1	SMO	71.55
2	LADTree	75.51

3	SMO	73.83

Figure 1. The LADTree algorithm illustrating the most significant words for classification into the positive or negative class.

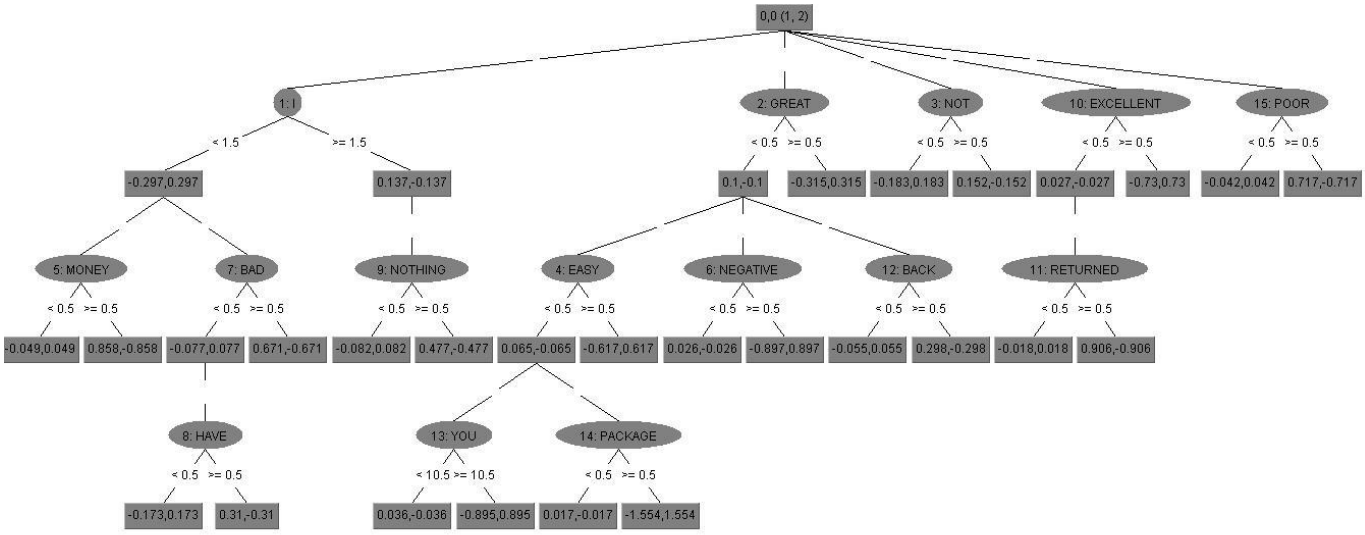

Figure 2. The agreement between the actual and predicted classification for Data 1. The upper curve (between the comment number 40 to 120) illustrates the predicted classification.

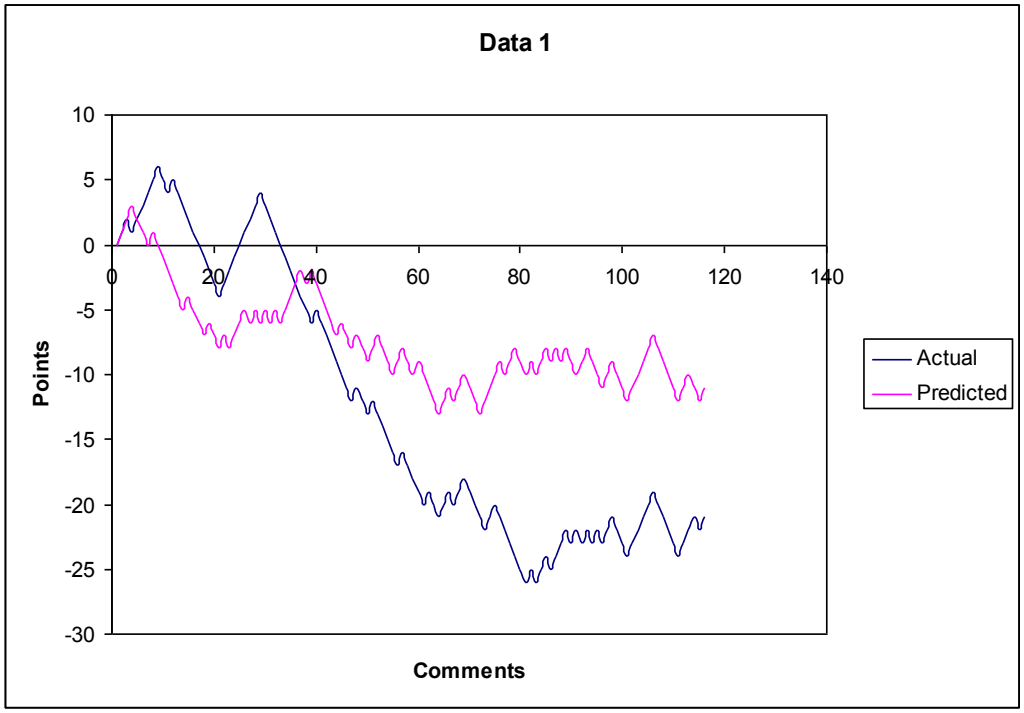

Figure 3.The agreement between the actual and predicted classification for Data 2. The upper curve (between the comment number 20 to 50) illustrates the predicted classification.

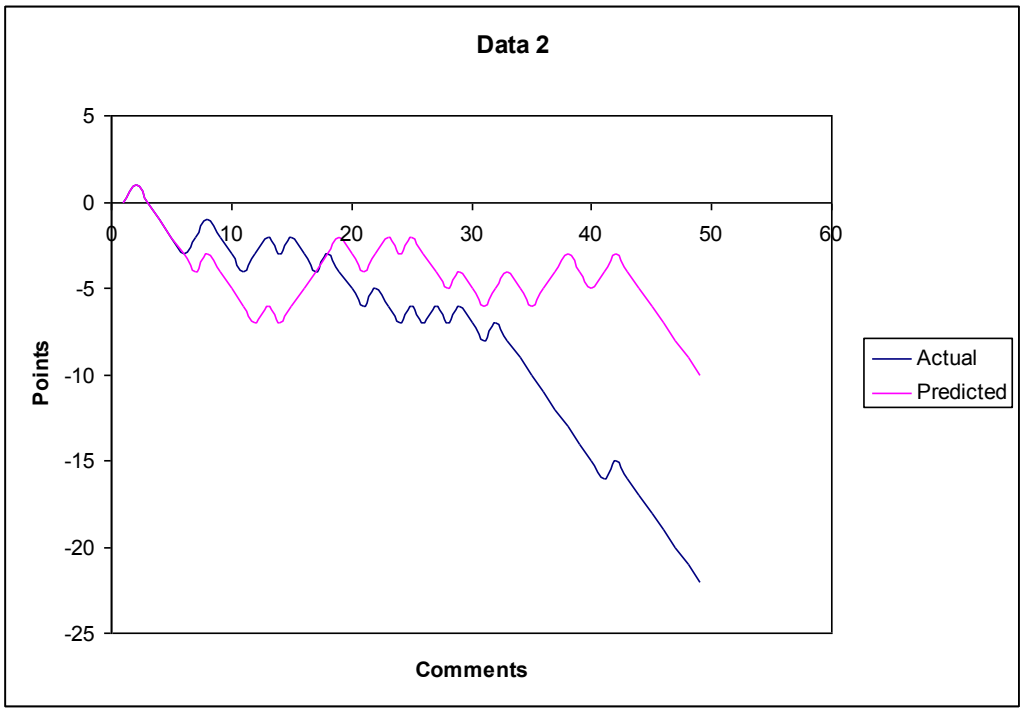

Figure 4.The agreement between the actual and predicted classification for Data 3. The lower curve illustrates the predicted classification.