Markov Chain Monte Carlo Methods and Evolutionary Algorithms for Automatic Feature Selection from Legal Documents

S. Pudaruth[1(✉)], K.M.S. Soyjaudah[2], and R.P. Gunputh[3]

[1] Department of Ocean Engineering and ICT, Faculty of Ocean Studies, University of Mauritius, Moka, Mauritius
s.pudaruth@uom.ac.mu
[2] Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Mauritius, Moka, Mauritius
s.soyjaudah@uom.ac.mu
[3] Department of Law, Faculty of Law and Management, University of Mauritius, Moka, Mauritius
rpgunput@uom.ac.mu

Abstract. In this paper, we present three different approaches for feature selection, starting from a naïve Markov Chain Monte Carlo random walk algorithm to more refined methods like simulated annealing and genetic algorithms. It is typical for textual data to have thousands of dimensions in their feature space which makes feature selection a crucial phase before the final classification. Classification of legal documents into eight categories was performed via a simple document similarity measure based on term frequency and the nearest neighbour concept. With an average success rate of 76.4%, the random walk algorithm not only performed better than the simulated annealing and genetic algorithms but also matched the accuracy of support vector machines. Although these methods have commonly been used for selecting appropriate features in other fields, their use in text categorisation have not been satisfactorily investigated. And, to our knowledge, this is the first work which investigates their use in the legal domain. This generic text classification framework can further be enhanced by using an active learning methodology for the selection of training samples rather than following a passive learning approach.

Keywords: Monte Carlo · Random walk · Genetic algorithm · Simulated annealing · Legal text categorisation · Court judgements

1 Introduction

Text categorisation or text classification is a sub-branch of text mining and natural language processing technique which attempts to assign documents to specific categories. Text categorisation is important in many applications such as the automated generation of metadata in document retrieval systems, question answer systems and search engines (Sahin 2007). Text documents are usually represented using the bag-of-words approach and vector space model. In this representation, a document or a

S.M. Thampi et al. (eds.), *Intelligent Systems Technologies and Applications*,
Advances in Intelligent Systems and Computing 683,
https://doi.org/10.1007/978-3-319-68385-0_12

document set can have thousands of vectors where each vector is usually a tuple consisting of a word and its term frequency. Because of the very high dimensions inherent in textual data, feature selection is one of the key steps in text categorisation.

Given a set of n words, it still possible to generate $2^n - 1$ subsets of 1 to n words and $\frac{n!}{r!(n-r)!}$ possible subsets of size r. Thus, there are more than 47 trillion ways of choosing 30 words form a bag of 50 words. So, for a given document set, it is usually not possible to consider all possible combinations of terms in order to find the best one. Therefore, when no exact formulation of a problem is possible, it is necessary to resort to approximations. In text mining, dimension reduction is usually achieved by converting all text to lowercase, removal of stopwords, stemming and lemmatisation. However, this is often not sufficient and the dimension often remains unusually large.

In this paper, we investigate three different methods which are all based on the principles of Markov chain Monte Carlo (MCMC) methods. These Monte Carlo techniques, due to their general applicability, has been used in a large variety of domains especially for studies involving simulations in physics, biology and computer science (Browne et al. 2012). Andrieu et al. (2003) wrote an interesting paper which bridges the gap between the Monte Carlo methods and traditional machine learning techniques. The basic Monte Carlo techniques and its various variance-reduction variants and their applications are described in detail. A theoretical foundation of the different techniques are also provided.

The use of simulated annealing and genetic algorithms as feature selection methods in the text mining field are scarce compared to other fields but not new. Genetic algorithms have been used in information retrieval systems by Gordon (1988) and Chen and Kim (1994). Although simulated annealing (Metropolis et al. 1953) is an older technique than genetic algorithm (Holland 1975), its use as a feature selection method in text classification is relatively recent (Wang et al. 2006; Yang et al. 2007).

In this work, inspired by the principles of Monte Carlo, we are able to show that by repeatedly determining the category of a document from a sample of features selected through a random walk or differential evolution, we are able to deduce the category of a court judgement to a higher accuracy using only a very small fraction of the total number of features. The use of the Monte Carlo approach allows a micro-local analysis to be performed on a high-dimensional problem. We believe that this new technique can become a tool of general applicability in the field of text classification.

The performance of machine learning classifiers hinges on the availability of large amounts of training data. Manual classification of court cases requires highly skilled professionals and is a time-consuming and costly undertaking. Through passive selection, we divided our dataset into several partitions of training set (10–90%) and testing set (90–10%). In the future, we intend to use the principles of active learning to optimise the choice of the training samples (Roy and McCallum 2001; Figueroa et al. 2012). Although support vector machines performs better on a 90–10 split, our Monte Carlo-based classification is able to achieve higher accuracies when the training set is less than 80% of the full dataset.

The remainder of this paper is organised as follows. Section 2 discusses on prior research on Markov Chain Monte Carlo methods, simulated annealing and genetic algorithms. Section 3 describes in detail how these methods have been adapted for

feature selection and text classification in our system. The experimental results and discussions are provided in Sect. 4. And finally, the conclusion is offered in Sect. 5 wherein some potential avenues for future research is indicated.

2 Related Works

For a modern and comprehensive literature review on the field of text document classification, the reader is referred to Khan et al. (2010). The paper starts with describing state-of-the-art applications of text mining and document classification. The pre-processing steps such as feature extraction, feature selection, dimensionality reduction and document representation are also explained. Machine learning techniques such as kNN, naïve Bayes, decision trees, support vector machines, artificial neural networks, fuzzy logic, genetic algorithms, classification rules and hybrid techniques have also been well tackled in reasonable depth.

The Monte Carlo method find its roots in the field of statistics in which random numbers are used to perform simulation (Liang and Wong 2000; Goncharov et al. 2007; Houghton et al. 2014). This method is often used in problems involving very high dimensions, for example, in the Travelling Salesman Problem where there is a very large number of potential solutions. The problem becomes rapidly intractable because of the exponential rise in the number of paths for each new node that is added to the existing network (Buxey 1979; Martin et al. 1991).

Monte Carlo simulation approaches has found wide applications in computer games (Browne et al. 2012). An algorithm known as the Monte-Carlo Tree Search Solver (MCTS-Solver) has been successfully applied in the popular game of Line of Action in order to find better strategies to play the game (Winands et al. 2008). It was shown that a program using the MCTS-Solver was able to win in 65% of matches.

An interesting application of the Monte Carlo Search Framework has been described in Branavan et al. (2012). In particular, they used a Monte Carlo approach to provide textual information retrieved from a manual to feed a game agent. This game-playing agent was able to outperform the uninformed and untrained agent (default AI) in the game of Civilization by a factor of 34%.

Another modern area where Monte Carlo simulations are being heavily used is in the field of bioinformatics. Ebbert et al. (2011) have used the Monte Carlo approach to generate samples for characterising uncertainty in multi-variate assays. This uncertainly information is very important for clinicians in order to properly classify different types of tumours. Diaconis (2009) has used the principles of Markov Chain Monte Carlo in order to decrypt messages. More recently, Monte Carlo methods have been applied in multi-label classification (Read et al. 2014).

The concept of an evolutionary Monte Carlo method is not new. Previously, it has been used in the scientific, engineering and mathematical filed for the estimation of various parameter values (Smith and Hussain 2012) and uncertainties in optimisation problems (Ter Braak 2006; Wu et al. 2006; Xiao 2007).

One of the earliest works which applied a genetic algorithm (GA) for the extraction of correlated terms was done by Desjardins et al. (2005). However, they concluded that the co-occurrences found by the GA did not improve the retrieval accuracy and more

work was required to understand the cognitive factors which would improve the relevancy of retrieved results.

Gavrilis et al. (2005) have used a GA for feature selection on 650 PUBMED abstracts spread into 5 categories. Using an SVM classifier, they achieved 85% accuracy using 20 features only. In another similar study on spam emails, they report an accuracy of 97%, again using only 20 features (Gavrilis et al. 2006).

A comparison was made between tf-idf (term frequency-inverse document frequency) and genetic algorithms by Khalessizadeh et al. (2006) for the identification of document topics in relatively short Persian texts. While precision values were very similar for both methods, GA did better than tf-idf on recall for all document sizes.

Pietramala et al. (2008) proposed the Olex-GA algorithm which assigns positive and negative rules to each feature (gene) in a chromosome. Their approach performed better on the OHSUMED dataset when compared with techniques such as naïve Bayes, C4.5, Ripper and Support Vector Machines. However, for the Reuters-21578 dataset, SVM did significantly better than Olex-GA.

Song and Park (2009) have used a genetic algorithm to find the optimal number of clusters in the Reuters-21578 dataset. The number of terms were reduced using latent semantic indexing (LSI) before the GA was applied. They were able to show that their algorithm performs better than previous methods. However, their study was based on only 1000 documents from 5 categories. Liu and Fu (2012) used an elitist GA to classify 100 web pages into 5 categories and obtained better performance than when using SVM with default parameters.

Chen et al. (2013) proposed a novel text classification procedure based on the chaos optimization theory and genetic algorithm. They tested their approach on the Reuters-21578 dataset and they were able to demonstrate that the algorithm requires a small feature set in order to provide comparable recall performance compared with earlier higher-dimensional techniques. Using GATE (Cunningham and Tablan 2002) and Weka (Hall et al. 2009), Rogers (2013) implemented a single pipeline for feature selection using genetic algorithms and classification several machine learning classifiers. Pavlyshenko (2014) have used a genetic algorithm to determine the optimal subset of keywords which could be used to identify an authors of English fiction texts.

A good introduction to feature selection in text mining using simulated annealing can be found in Bagheri et al. (2014). They demonstrated that their proposed approach delivered similar performance to chi-squared when tested on a Persian dataset consisting of 7 categories. There were about 800 documents in each category. Zhu et al. (2015) further showed that an improved simulated annealing algorithm (SAA) can select better features than information gain (IG), mutual information (MI) and chi-squared (CHI).

Moshki et al. (2015) tested an extended version of the simulated algorithm (SAGRASP) on a diverse set of data and showed that it was able to do better than FCGRASP (Bermejo et al. 2011). Zhu et al. (2009) successfully combined the genetic algorithm and simulated annealing (SA) to extract protein sequences using OpenMP. They reported that their proposed solution did not get trapped in local maxima and could find global optima faster. Two decades earlier, Esbensen and Mazumder (1994) used a mixture of SA and GA, which they called SAGA, to determine an optimal placement for macro-cells.

3 Description of Algorithms

3.1 Markov Chain Monte Carlo Random Walk (MCMCRW)

A random walk is a random or stochastic process that may consist of a series of random steps (Pemantle 2007; Samad 2013). The principles of random walk has found wide applications in different fields of computer science such as the analysis of computer networks (Zhong et al. 2008), computer security (Zhou 2016), bioinformatics (Draminski et al. 2008) and text classification (Hassan et al. 2007).

In our system, a random walk consists of a sequence of similar operations in which a subset of k elements are randomly selected (with replacement) from a list of n elements from each of the m categories, p number of times. In computer science, this is known as a Markov Chain Monte Carlo (MCMC) process. Each element is a word and these $m * n$ elements (mainlist) are initially selected using term frequency. We have two sets of data: the training set and the testing set. The training set is only used for the extraction of representative elements for each categogy. We do not use a wrapper-style classifier (Jovic et al. 2015) to measure the classifier accuracy, instead we use a naïve classifier based on term frequency. Each of the m sublists is compared to every document in the testing set and the sum of all the k words is computed. The sublist with the highest score is taken as the predicted category. This classifier can be considered as a simplified version of the k-nearest neighbour classifier. A simple majority voting is then carried out on the results obtained after the p iterations.

3.2 Boosted Simulated Annealing (BSA)

The basic principles in the simulated annealing algorithm was described by Metropolis et al. (1953). Their aim was to simulate the movement of atoms at a finite temperature. In 1970, Hastings showed how this method could be generalised to solve problems in statistics. Kirkpatrick et al. (1983) took up the same basic ideas but added the concepts of high and low temperatures and compared the algorithm to the annealing process in metals, from which the algorithm got its name. It is only very recently that the simulated annealing algorithm has been used for feature selection in the text classification field (Wang et al. 2006; Yang et al. 2007; Bagheri et al. 2014; Zhu et al. 2015; Moshki et al. 2015).

In our system, we implemented the standard simulated annealing algorithm but is boosted with a good initial sample which is produced by random sampling through a random walk of 100 steps. The selection of elements is similar to MCMCRW. However, in BSA, the next step is not independent on the current one. The next list is generated by replacing t elements in the current best list by t other elements from each category (sublist) from the mainlist. The number t is a temperature variable which decreases steadily to 0 from the first iteration until the last one. If the accuracy increases after this small change, the best list is updated. However, even if the accuracy decreases by x percent, we still consider this new list as the current one. If the accuracy decreases by x percent or more, the current list is discarded and a new one is generated. This entire process is repeated q times. Our algorithm is also stateful in that it has a memory to store the best list from any of the q iterations.

3.3 Genetic Algorithms

A genetic algorithm (GA) is often described as a meta-heuristic algorithm which emulates the Darwinian's theory of natural evolution through the biological processes of selection, mating and mutation (Mitchell 1998). Since its formulation in 1970 by Holland, GAs has found wide applications, not only in scientific areas but also in business applications. Thomas and Sycara (2002) have used GAs for predicting stock prices while Borg (2009) has used GAs for the automatic extraction of definitions. In combination with other methods, Waad et al. (2014) used a genetic algorithm to select the best features to assess the credit worthiness of a potential client. A recent and novel application of GA is in the detection of errors in SQL instructions (Moncao et al. 2013). Also, as stated earlier, GAs have also been used in the information retrieval domain since more than two decades (Atkinson-Abutridy et al. 2004; Al-Maqaleh et al. 2012).

In our system, we maintain a population of r mainlist. In GA's jargon, a mainlist can be considered as a chromosome. A mainlist is a list of sublists and each sublist contains the most frequent terms in one category of documents. As mentioned earlier, each term is a word (gene). A fitness score (classification accuracy) is calculated for each of these r lists. The best b lists (chromosomes) are selected in each iteration for crossover and mutation. Each of these b lists are randomly paired with each other (without duplication) to generate $\frac{b}{2}$ pairs and c elements are then chosen randomly from one member in each pair and are swapped. This is the crossover operation whereby b new lists are created and the previous b lists are discarded*. Our crossover operation is slightly different from previous approaches that use fixed locations for genes in that we do not exchange a segment of the chromosome, instead, we exchange genes selected randomly from anywhere in the chromosome. The rationale for this heuristic approach is that the genes (words) are independent and this reduces the likelihood of collisions. The next operation is mutation which is achieved by the substitution of d elements in each list by new elements from the mainlists. The remaining r-b lists are rejected and new ones are generated randomly in order to keep the size of the population constant. These processes are repeated q times. The best list from each of these q iterations are stored in a separate memory*.

4 Experiments and Settings

4.1 Experimental Corpus

Our dataset consists of 294 judgements which were delivered in the Supreme Court of the Republic of Mauritius in the year 2013. The cases have previously been classified manually into eight categories: homicide, road traffic offences, drugs, other criminal offences, company law, labour law, land law and contract law. It is the same dataset that we have used previously in an earlier work (Pudaruth et al. 2016).

4.2 Document Pre-processing

Because textual data is inherently noisy, it is important to filter out those elements that would negatively impact on the performance of classifiers. Thus, all the data was first converted to lowercase after which all digits, symbols, short words and stopwords were filtered out. Besides the common English stopwords, the list also included words from the legal domain such as case, act, section, law, court, appellant, respondent, judge and many others. These words tend to occur in almost all judgements and their frequencies are also very high. This is an interesting issue because the same words could have been strong differentiators if our objective was to look for legal documents in a mixture of documents from other domains. With the help of Wordnet (2017), all non-English words and all verbs were removed from the dataset. All the pre-processing steps taken together reduced the feature set from 6048 to 2485 words.

4.3 Experimental Environment and Algorithmic Parameters

The documents are kept in a parent folder with eight directories. Each directory contains the files for one specific category whereby each judgement is stored as a separate textfile. The software for document pre-processing and feature selection have been implemented in Python 2.7.12 (Spyder 3.0.0) from the Anaconda distribution. The scikit-learn library for Python has been used for the machine learning part. A computer running the 64-bit Windows 7 Professional N (SP1) operating system has been used in this study. The processor is an Intel(R) Core(TM) i5-4200 M CPU running at 2.5 GHz on 8.00 GB of RAM on a hard disk of 650 GB.

The number of iterations for each of the 3 Monte Carlo methods was 100. A sample consisted of 30 (k) elements drawn from a larger set of 100 (n) most frequent words from each of the 8 (m) categories. The mutation rate for both the simulated annealing and genetic algorithms was 20%. This means that for every 30 elements, 6 elements were exchanged. The initial temperature for SA was set at 10 (for every sublist) and this was reduced by 0.1 after every iteration until it reached a minimum value of 1. The SA algorithm was allowed to accept solutions which was 5% (x) worse than the current one in an attempt to avoid local maxima. The crossover rate for the GA was also set at 20%. The parameters were chosen empirically, after conducting a large set of experiments and observing their impact on the accuracy. All the algorithms were run 5 times on each split percentage and an average was made.

4.4 Experimental Results

In this sub-section, we present the detailed results to demonstrate the effectiveness of the Markov Chain Monte Carlo (MCMC) Random Walk algorithm compared to simulated annealing (SA), genetic annealing (GA) and support vector machines (SVM). The dataset has been split into nine different sets of training and testing data, as shown in Table 1, with a view to understand the influence of different training sizes on feature selection and classification accuracy. A comparison with SVM is also provided.

Table 1. Details of cases dataset

Categories	Code	No. of Cases
Company Law	Comp	22
Contract Law	Cont	56
Other Criminal Offences	Crim	48
Drugs	Drugs	44
Homicide	Homi	14
Labour Law	Labo	17
Land Law	Land	55
Road Traffic Offences	Road	38

In general, classification accuracy increases when the size of the training set increases, as shown in Table 2. For all training sizes, MCMCRW and SVM are more effective than SBA and GA. The best accuracy of 83% is obtained by MCMCRW at 70% of the training set and by SVM at 90% of the training set. When the training size is 20% or less, all the three algorithms (MCMCRW, BSA and GA) does better than SVM (with default settings and parameters). When the training size is between 40 and 70%, only MCMCRW is able to outperform SVM. The results illustrate that our random walk algorithm can produce satisfactory results even with low amount of training data. In the legal field where it is very costly and time-consuming to produce annotated data as this has to be done by highly trained professionals, the random walk algorithm only requires a few relevant documents for training for each category to deliver acceptable results.

Table 2. Classification accuracy v/s training/testing size

Training Set (%)	10	20	30	40	50	60	70	80	90	Average
Testing Set (%)	90	80	70	60	50	40	30	20	10	
MCMC Random Walk	68	76	78	70	76	80	**83**	78	79	76.4
Boosted Simulated Annealing (BSA)	64	63	63	61	66	65	69	74	**76**	66.8
Genetic Algorithm (GA)	62	66	64	60	66	64	72	73	**79**	67.3
Support Vector Machines (SVM)	53	54	63	67	73	76	75	78	**83**	69.1

Table 3 shows the detailed results for one run on a split of 70/30, in which there were 201 cases in the training set and 93 cases in the testing set. The overall accuracy for this run was 84%. Accuracy is defined as the number of correctly classified documents over the total number of documents in the testing set. The *Road traffic offences* category has a prefect recall, which means that we have been able to retrieve all instances of this category from the testing set, while the *Contract* category has the lowest recall because 3 of its cases have been incorrectly retrieved as a *Labour* case and another two as a *Land* case. However, these misclassifications are quite comprehensible as these 3 categories share many terms in common as they all deal with contractual issues.

Table 3. Confusion matrix

Category	Comp	Cont	Crim	Drug	Homi	Labo	Land	Road	Total	Recall
Comp	6	0	0	0	0	0	1	0	7	0.86
Cont	0	12	0	0	0	3	2	0	17	**0.71**
Crim	0	0	13	1	0	0	0	1	15	0.87
Drug	0	0	2	12	0	0	0	0	14	0.86
Homi	0	0	0	1	4	0	0	0	5	0.80
Labo	1	0	0	0	0	5	0	0	6	0.83
Land	2	1	0	0	0	0	14	0	17	0.82
Road	0	0	0	0	0	0	0	12	12	**1.00**
Total	9	13	15	14	4	8	17	13	93	0.82
Precision	0.67	0.92	0.87	0.86	**1.00**	**0.63**	0.82	0.92	0.84	

The *Homicide* category has the highest precision followed closely by *Road traffic offences* and *Contract*. The *Labour* and *Company* categories have the lowest precision values. Nine documents have been returned as belonging to the *Company* category, however, only six of them are actually company law cases. One document belongs to the labour law category while another two belong to the land law category. Again, we see that there is some overlap in features between the *Contract*, *Company* and *Labour* categories. Some additional work will be necessary in order to reduce this mix-up.

The Olex-GA system proposed by Pietramala (Pietramala et al. 2008) did slightly better than SVM on the OHSUMED dataset but less well on the Reuters-21578 dataset. The SA algorithm proposed by Yang et al. (2007) performed slightly better than kNN on some settings. Wang et al. (2006) reported a similar result. However, it is always very difficult to offer a fair comparison when comparing GAs and SAs with machine learning classifiers. The variation in the total number of documents in the dataset, the number of classes, the size of the documents, the number of training & testing samples, the nature & complexity of the documents, the multitude of parameters used in the evolutionary algorithms and in the classifiers all lead to a very difficult comparison between the various studies.

5 Conclusions

This paper presents three different feature selection methods and a general text categorisation framework. After an extensive empirical evaluation with a set of 294 Supreme Court judgements spread into eight areas of law, we found that the simulated annealing and genetic algorithms, with an average classification accuracy of about 67%, did less well than the conceptually simpler random walk while the latter's performance was on average better than support vector machines. The idea of using evolutionary operations for the selection of suitable features is not new, however, full-fledged implementation of these techniques in the domain of text classification is relatively recent. The random walk algorithm is very robust as it does not fluctuate as much as machine learning classifiers do when the number of training samples is

reduced. The added benefit of this system is its simplicity and understandability. It is very easy for a user to improve the quality of the process by adding new training samples or new filters. In future work, we intend to choose the training instances using an active learning technique instead of passive learning. Combining the strengths of each of these feature selection methods into a single algorithm is also a potential avenue for further research.

References

Al-Maqaleh, B.M., Shahbazkia, H.: A genetic algorithm for discovering classification rules in data mining. Int. J. Comput. Appl. **41**(18), 40–44 (2012)

Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Mach. Learn. **50**, 5–43 (2003)

Atkinson-Abutridy, J., Mellish, C., Aitken, S.: Combining information extraction with genetic algorithms for text mining. IEEE Intell. Syst. **19**(3), 22–30 (2004)

Bagheri, A., Saraee, M., Nadi, S.: PSA: a hybrid feature selection approach for Persian text classification. J. Comput. Secur. **1**(4), 261–272 (2014)

Bermejo, P., Gamez, J.A., Puerta, J.M.: A GRASP algorithm for fast hybrid filter-wrapper feature subset selection in high-dimensional datasets. Pattern Recogn. Lett. **32**(5), 701–711 (2011)

Borg, C.: Automatic Definition Extraction using Evolutionary Algorithms. Thesis (MSc), University of Malta, Malta (2009)

Branavan, S.R.K., Silver, D., Barzilay, R.: Learning to win by reading manuals in a Monte Carlo framework. J. Artif. Intell. Res. **43**, 661–704 (2012)

Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of Monte Carlo tree search methods. IEEE Trans. Comput. Intell. AI Games **4**(1), 1–43 (2012)

Buxey, G.M.: The vehicle scheduling problem and Monte Carlo simulation. J. Oper. Res. Soc. **30**(6), 563–573 (1979)

Chen, H., Kim, J.: GANNET: a machine learning approach to document retrieval. J. Manag. Inf. Syst. **11**(3), 7–41 (1994)

Chen, H., Jiang, W., Li, C., Li, R.: A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. Math. Problems Eng. 2013, Article ID: 524017

Cunningham, M., Tablan, B.: GATE: a framework and graphical development environment for robust NLP Tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), 7–12 July 2002, Philadelphia, Pennsylvania (2002)

Desjardins, G., Godin, R., Proulx, R.: A genetic algorithm for text mining. WIT Trans. Inf. Commun. Technol. **35**, 133–142 (2005)

Diaconis, P.: The Markov chain Monte Carlo revolution. Bull. Am. Math. Soc. **46**, 179–205 (2009)

Draminski, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J., Komorowski, J.: Monte Carlo feature selection for supervised classification. Bioinformatics **24**(1), 110–117 (2008)

Ebbert, M.T.W., Bastien, R.R.L., Boucher, K.M., Martin, M., Carrasco, E., Caballero, R., Stijleman, I.J., Bernard, P.S., Facelli, J.C.: Characterization of uncertainty in the classification of multivariate assays: application to PAM50 centroid-based genomic predictors for breast cancer treatment plans. J. Clin. Bioinform. **1**, 37 (2011)

Esbensen, H., Mazumder, P.: SAGA: a unification of the genetic algorithm with simulated annealing and its application to macro-cell placement. In: Proceedings of the 7th International Conference on VLSI Design, Calcutta, India, 5–8 January 1994, pp. 211–214 (1994)

Figueroa, R.L., Zeng-Treitler, Q., Ngo, L.H., Goryachev, S., Wiechmann, E.P.: Active learning for clinical text classification: is it better than random sampling? J. Am. Med. Inform. Assoc. **19**(5), 809–816 (2012)

Gavrilis, D., Tsoulos, I.G., Dermatas, E.: Stochastic classification of scientific abstracts. In: Proceedings of the 6th Speech and Computer Conference, Patras, Greece (2005)

Gavrilis, D., Tsoulos, I.G., Dermatas, E.: Neural recognition and genetic features selection for robust detection of E-mail spam. Adv. Artif. Intell. **3955**, 498–501 (2006)

Goncharov, Y., Okten, G., Shah, M.: Computation of the endogenous mortgage rates with randomized quasi-Monte Carlo simulations. Math. Comput. Model. **46**(3–4), 459–481 (2007)

Gordon, M.: Probabilistic and genetic algorithms for document retrieval. Commun. ACM **31**(10), 1208–1218 (1988)

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. **11**(1), 10–18 (2009)

Hassan, S., Mihalcea, R., Banea, C.: Random walk term weighting for improved text classification. Int. J. Semant. Comput. **1**(4), 421–439 (2007)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Michigan (1975)

Houghton, J., Siegel, M., Wirsch, A., Moulton, A., Madnick, S., Goldsmith, D.: A survey of methods for data inclusion in system dynamics models: methods, tools and applications. Massachusetts Institute of Technology, Cambridge, Working Paper CISL# 2013-03 (2014)

Jovic, A., Brkic, K., Bogunovic, N.: A review of feature selection methods with applications. In: Proceedings of the 38th IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2015), Opatija, Croatia, 25–29 May 2015, pp. 1200–1205 (2015)

Khalessizadeh, S.M., Zaefarian, R., Nasseri, S.H., Ardil, E.: Genetic mining: using genetic algorithm for topic based on concept distribution. In: Proceedings of the World Academy of Science, Engineering and Technology (2006)

Khan, A., Baharudin, B., Lee, L., Khan, K.: A review of machine learning algorithms for text documents classification. J. Adv. Inf. Technol. **1**(1), 4–20 (2010)

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220** (4598), 671–680 (1983)

Liang, F., Wong, W.H.: Evolutionary Monte Carlo: applications to Cp model sampling and change point problem. Stat. Sin. **10**, 317–342 (2000)

Liu, X., Fu, H.: A hybrid algorithm for text classification problem. Electrical review, R. 88 NR 1b (2012)

Martin, O., Otto, S.W., Felten, E.W.: Large-step Markov chains for the travelling salesman problem, p. 16. CSETech, Paper (1991)

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)

Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1998)

Moncao, A.C.L., Camilo-JR, C.G., Queiroz, L.T., Rodrigues, C.L., Leitao-JR, P.S., Vincenzi, A. M.R.: Applying genetic algorithms to data selection for SQL mutation analysis. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO 2013), Amsterdam, The Netherlands, 7–10 July 2013, pp. 207–208 (2013)

Moshki, M., Kabiri, P., Mohebalhojeh, A.: Scalable feature selection in high-dimensional data based on GRASP. Appl. Artif. Intell. **29**, 283–296 (2015)

Pavlyshenko, B.: Genetic optimization of keywords subset in the classification analysis of texts authorship. J. Quant. Linguist. **21**(4), 341–349 (2014)

Pemantle, R.: A survey of random processes with reinforcement *. Prob. Surv. **4**, 1–79 (2007)

Pietramala, A., Policicchio, V.L., Rullo, P., Sidhu, I.: A genetic algorithm for text classification rule induction. Lect. Notes Comput. Sci. **5212**, 188–203 (2008)

Pudaruth, S., Soyjaudah, K.M.S., Gunputh, R.P.: Categorisation of supreme court cases using multiple horizontal thesauri. Intell. Syst. Technol. Appl. **2**, 355–368 (2016)

Read, J., Martino, L., Luengo, D.: Efficient Monte Carlo methods for multi-dimensional learning with classifier chains. Pattern Recogn. **47**, 1535–1546 (2014)

Rogers, B.C.: Using genetic algorithms for feature set selection in text mining. Thesis (MSc), Miami University, Oxford, Ohio (2013)

Roy, N., Mccallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 441–448 (2001)

Sahin, I.E.: Online text categorization using genetic algorithms. Bilkent University, Turkey, Technical report, BU-CE-0704 (2007)

Samad, S.A.: Random walk oversampling technique for minority class classification. Thesis (MSc), Tampere University of Technology (2013)

Smith, R., Hussain, M.S.: Genetic algorithm sequential Monte Carlo methods for stochastic volatility and parameter estimation. In: Proceedings of the World Congress on Engineering (WCE 2012), London, UK, 4–6 July 2012, vol. 1 (2012)

Song, W., Park, S.C.: Genetic algorithm for text clustering based on latent semantic indexing. Comput. Math Appl. **57**, 1901–1907 (2009)

ter Braak, C.J.F.: A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. Stat. Comput. **16**(3), 239–249 (2006)

Thomas, J.D., Sycara, K.: Integrating genetic algorithms and text learning for financial prediction. In: Proceedings of the Genetic and Evolutionary Computing Conference (GECCO), Las Vegas, Nevada, pp. 72–75

Waad, B., Mufti, G.B, Liman, M.: A new feature selection technique applied to credit scoring data using a ranked aggregation approach based on: optimisation, genetic algorithm and similarity. In: Osei-Bryson, K., Barclay, C. (eds.) Knowledge Discovery Process And Methods To Enhance Organisational Performance, pp. 347–376. CRC Press, Boca Raton (2014)

Wang, R., Youssef, A.M., Elhakeem, A.K.: On some feature selection strategies for spam filter design. In: Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2006), Ottawa, Canada, 7–10 May 2006, pp. 2155–2158 (2006)

Winands, M.H.M., Bjornsson, Y., Saito, J.T.: Monte Carlo tree search solver. In: Proceedings of the 6th International Conference on Computers and Games, pp. 25–36 (2008)

WordNet: a lexical database for English. Princeton University (2017). https://wordnet.princeton.edu/wordnet/. Accessed 31 Jan 2017

Wu, J., Zheng, C., Chien, C.C., Zheng, L.: A comparative study of Monte Carlo simple genetic algorithm and noisy genetic algorithm for cost-effective sampling network design under uncertainty. Adv. Water Resour. **29**, 899–911 (2006)

Xiao, X.: Advanced Monte Carlo techniques: an approach for foreign exchange derivative pricing. Thesis (PhD), University of Manchester, UK (2007)

Yang, C., Li, Y., Zhang, C., Hu, Y.: A fast KNN algorithm based on simulated annealing. In: Proceedings of the International Conference on Data Mining, Las Vegas, Nevada, 25–28 June 2007, pp. 46–51 (2007)

Zhong, M., Shen, K., Seiferas, J.: The convergence-guaranteed random walk and its application in peer-to-peer networks. IEEE Trans. Comput. **57**(5), 619–633 (2008)

Zhou, Y.: A random-walk based privacy-preserving access control for online social networks. Int. J. Adv. Comput. Sci. Appl. **7**(2), 74–79 (2016)

Zhu, F., Li, H., Yao, N., Zhu, H.: Text feature selection applied by improved SAA*. J. Comput. Inf. Syst. **11**(17), 6419–6427 (2015)

Zhu, H., Chen S., Pu, C., Liu, Y., Eguchi, K., Zhang, S.: Paralleling genetic annealing algorithm with OpenMP. In: Proceedings of the 2nd IEEE International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2009), Tianjin, China, 1–3 November 2009