Modern Applications of Machine Learning

George Tzanis, Ioannis Katakis, Ioannis Partalas, Ioannis Vlahavas

Department of Informatics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece, {gtzanis, katak, partalas, vlahavas}@csd.auth.gr

Machine learning is one of the older areas of artificial intelligence and concerns the study of computational methods for the discovery of new knowledge and for the management of existing knowledge. Machine learning methods have been applied to various application domains. However, in the few last years due to various technological advances and research efforts (e.g. completion of the Human Genome Project, evolution of the Web), new data have been available and consequently new domains where machine learning can be applied have been arisen. Some of these modern applications are learning from biological sequences, learning from email data, and learning in complex environments such as Web. In this paper we present the above three application domains as well as some recent efforts, where machine learning techniques are applied in order to analyze the data provided by these domains.

Keywords
Bioinformatics, Learning from Email, Machine Learning, Reinforcement Learning

1. Introduction

A cognitive system tries to understand the concepts of its environment by using a simplified interpretation of this environment called *model*. The procedure of constructing such a model is called *inductive learning*. Moreover, a cognitive system is able to organize its experience by constructing new structures called *patterns*. The construction of models and patterns by a cognitive system using a dataset is called *machine learning*. Machine learning tasks can be classified into the following two groups:

- Supervised learning
- Unsupervised learning

A model describes the whole set of data and is also characterized as *predictive model* since it can be used to predict the output of a function (*target function*) for a given value in the function's domain. Moreover, a model provides some qualitative information about the data. In contrast, a pattern describes only a portion of the data and is characterized as *informative pattern*.

1.1 Supervised Learning

This kind of learning is also known as *learning from examples*. In supervised learning the cognitive system has to learn a concept or a function that is actually the description of a model. In particular the system is provided with a set of examples. The output of the target function for each of these examples is also available. The system has to discover the description of the model based on the output of the function. For evaluation purposes a model is built using a subset of the data (*training set*), while the remaining data are used to evaluate the constructed model (*test set*).

Two learning tasks are recognized in supervised machine learning, namely *classification* and *regression*. Classification concerns the construction of prediction models for functions with discrete range, while regression concerns the construction of prediction models for functions with continuous range. The most common supervised machine learning methods are the following:

- *Concept Learning.* The cognitive system is provided with examples that belong (positive examples) or do not belong (negative examples) in a concept (class). Then, the system is called to

produce a generalized description of the concept in order to decide for future cases based on this description.

- *Classification or Decision Tree Induction.* Classification or decision tree induction methods are very popular and are used for the approximation of discrete target functions. These methods construct tree structures that represent graphically the training data. The main advantage of decision trees is that they are easily interpreted. Decision trees can also be represented as "*if-then*" rules.
- *Rule Learning.* Rule learning includes the induction of "*if-then*" rules, called *classification rules*. Classification rules are used for the approximation of discrete target functions.
- *Instance Based Learning*. In this kind of learning the data are stored in their raw format. When the system is called to decide for a new case it examines the relationship between the new case and each of the stored examples. This kind of learning is also known as *lazy learning*, since the learning process is deferred until a new case appears.
- *Bayesian Learning*. This kind of learning is based on Bayes theorem and includes methods that utilize probabilities. Existing knowledge can be incorporated in the form of initial probabilities.
- *Linear Regression*. Linear regression is a method for describing a target function with a linear combination of a number of other variables. The target function's range must be a continuous interval.
- *Neural Networks*. Neural networks can be used both for classification and regression. Their function is based on biological patterns and various procedures that simulate the human brain's activity are used.
- *Support Vector Machines (SVMs)*. SVMs are based on statistical learning theory and neural networks. SVMs are very popular in classification and regression tasks and usually present very good prediction accuracy.

1.2 Unsupervised Learning

This kind of learning is also known as *learning from observation*. In unsupervised learning the system has to discover any patterns (i.e. associations or clusters) based only on the common properties of the example without knowing how many or even if there are any patterns. The main unsupervised machine learning methods are the following:

- *Association Rule Mining.* Association rule mining emerged in 1993 [1] and has many contributions from the research area of databases. It was introduced as a market basket analysis method. Association rules are implications of the form $A \Rightarrow B$. The interpretation of the above rule is that when item A appears in a basket, then item B will also appear in the same basket.
- *Sequential Pattern Mining.* Sequential pattern mining concerns the learning from ordered data. The order is usually temporal. It has also many contributions from the research area of databases and has been proposed [2] as an extension of association rules mining.
- *Clustering.* Clustering is the procedure of discovering clusters of examples, so that examples that belong to the same cluster are as similar as possible, while examples belonging in separate clusters are as dissimilar as possible.

1.3 Applications

Machine learning has been extensively applied in various application domains. Some of the most popular applications include medical diagnosis, credit risk analysis, customer profiling, market segmentation, targeted marketing, retail management and fraud detection. The last years due to various technological advances and research efforts like the completion of the Human Genome Project (http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml) and the evolution of the Web, new data have been available and consequently new domains where machine learning can be applied have been arisen. Some of these modern applications are learning from biological sequences, learning

from text, and learning in complex environments such as Web. The above three modern applications of machine learning are presented below. In particular, in section 2 the application of machine learning on biological sequences is presented, section 3 deals with learning from text and section 4 concerns focused crawling using reinforcement learning. Finally, section 5 concludes the paper.

2. Learning from Biological Sequences

The rapid progress of computer science in the last decades has been closely followed by a similar progress in molecular biology. Undoubtedly, the use of computational tools has given a boost in the collection and analysis of biological data, creating one of the hottest areas of research, namely *bioinformatics*. Biological sequences represent a large portion of the biological data that require the use of computational tools in order to be analyzed. The large size of the sequences and the numerous possible features are the main reasons behind the urgent need methods that allow for the efficient analysis of such data and the delivery of accurate and reliable knowledge to the domain expert. The field of machine learning provides the biologists with a big set of tools for the analysis of these data.

Many machine learning techniques have been proposed to deal with the identification of specific biological sequence segments. The most common include neural networks, Bayesian classifiers, decision trees, and Support Vector Machines [3, 4, 5]. Sequence recognition algorithms exhibit performance tradeoffs between increasing sensitivity (ability to detect true positives) and decreasing selectivity (ability to exclude false positives) [6]. However, as Li et al. [7] state, traditional machine learning techniques cannot be directly applied to this type of recognition problems. Thus, there is the need to adapt the existing techniques to this kind of problems. Attempts to overcome this problem have been made using feature generation and feature selection [7, 8]. Another machine learning application is the use of clustering algorithms to group structurally related biological sequences.

2.1 Prediction of Translation Initiation Sites

The main structural and functional molecules of an organism's cell are *proteins*. The information concerning the synthesis of each protein is encoded by the genetic material of the organism. The genetic material of almost every living organism is *DNA*. Another molecule that plays an important role in protein synthesis is *RNA*. DNA and RNA belong to a family of molecules called *nucleic acids*. Both proteins and nucleic acids are sequences of smaller molecules, *amino acids* and *nucleotides* respectively. A sequence can be represented as a string of different symbols. There are twenty amino acids and five nucleotides. Every nucleotide is characterized by one of the following letters: A, C, G, T, U. DNA may contain a combination of A, C, G, and T. In RNA U appears instead of T. Proteins are synthesized by the following process. DNA is used as template for the synthesis of RNA (*transcription*). Then RNA is used as template for the synthesis of a protein molecule (*translation*).

Translation takes place by an organelle called *ribosome*. The mRNA sequence is scanned by the ribosome, which reads triplets, or *codons*, of nucleotides and "translates" them into amino acids. Thus, a protein consisting of n amino acids is encoded by a sequence of $3n$ nucleotides. Since there are 64 different triplets formed from an alphabet of four nucleotides and the total number of amino acids is 20, it is obvious that some amino acids are encoded by more than one codon. Moreover, the triplet AUG, that encodes amino acid methionine is also used as a translation initiation codon. Finally, there are three stop codons for the termination of translation (UAG, UAA and UGA).

Translation, usually, initiates at the AUG codon nearest to the start of the RNA sequence. However this is not always the case, since there are some escape mechanisms that allow the initiation of translation at following AUG codons. Due to these mechanisms the recognition of the Translation Initiation Site (TIS) on a given sequence becomes more difficult. After the initiation of translation, the ribosome moves along the RNA molecule, towards the end of the sequence and reads the next codon. This process is repeated until the ribosome reaches a stop codon. For each codon read, the proper amino acid is brought to the protein synthesis site and is joined to the protein chain, which by this way is elongated.

The recognition of the TIS is essential for better understanding of the process of translation. It has been recognized as one of the most critical problems in molecular biology that requires the generation of classification models, in order to accurately and reliably distinguish the valid TISs from a set of false ones.

Although many approaches have been proposed to deal with this problem, there is a great potential for the improvement of their accuracy. In [10] we apply machine learning methods to tackle the problem of the prediction of TISs in DNA sequences. We use a large number of features and different algorithms in order to build more accurate models. Some of the features are directly extracted from the raw sequences, concerning the nucleotides present at each position of the sequence, but most of them are generated. Along with the features already discussed in other papers [5, 8], we have generated and proposed the use of some new ones (*up-down_x*, *up_pos_k_x*, *down_pos_k_x* in Table 1). We have shown that a combination of these features improves the accuracy of the prediction models. In [11] we have presented an extension of [10], where a step of grouping the sequences according to criteria based on the sequence length was incorporated. Moreover, instead of nucleotide pattern, amino acid patterns were used as features. Finally, a multiple classifier system was built. For our experiments we used a real world dataset that contains processed DNA sequences collected from vertebrate organisms [12].

In the following lines we describe the approach we have followed in [11] in order to construct a multiple classifier system for the prediction of TISs in genomic sequences. Our approach consists of a number of steps. Each of these steps is described in detail in the following lines.

Step 1: All sequences are scanned and every candidate TIS is detected.

Step 2: The candidate TISs found in step 1 are grouped according to the length of the sequence before the AUG codon (upstream) and after the AUG codon (downstream). By this way the initial dataset of candidate TISs is divided into a number of smaller datasets. In our setup we have divided the initial dataset in 4 smaller datasets (This step was absent from our approach in [10])

Step 3: For each of the candidate TISs the value of a number of features is calculated. More details about these features are listed in Table 1.

Step 4: The features are evaluated among the instances of every group according to their impact in the accuracy of classification. In our setup we have used the information gain measure.

Step 5: A number of the top ranked features is selected and a classifier is built for each of the data subsets.

Table 1 The features used in our approach [11].

Feature	**Description**
up_*x*	Counts the number of occurrences of amino acid *x* in the upstream region
down_*x*	Counts the number of occurrences of amino acid *x* in the downstream region
up-down_*x*	Counts the difference between the number of occurrences of amino acid *x* in the upstream region and the number of occurrences of amino acid *x* in the downstream region
up_pos_*k*_*x*	Counts the number of occurrences of nucleotide *x* in the k^{th} position of the upstream codons (k $\in \{1, 2, 3\}$)
down_pos_*k*_*x*	Counts the number of occurrences of nucleotide *x* in the k^{th} position of the downstream codons (k $\in \{1, 2, 3\}$)
up_-3_[AG]	A Boolean feature that is true if there is an A or a G nucleotide three positions before the AUG codon, according to Kozak's pattern (GCC[**AG**]CCaug**G**) [9]
down_+1_G	A Boolean feature that is true if there is a G nucleotide in the first position after the AUG codon, according to Kozak's pattern (GCC[**AG**]CCaug**G**)
up_AUG	A Boolean feature that is true if there is an in-frame upstream AUG codon
down_stop	A Boolean feature that is true if there is an in-frame downstream stop codon

Finally, a new instance, namely a new candidate ATG, is assigned to one of the groups according to the length of its upstream and downstream regions' length and is classified by the corresponding classifier.

Our approach has been tested using various classification algorithms (e.g. C4.5, Naïve Bayes, PART, RIPPER) and presented better classification accuracy than other approaches.

3. Learning from Email Data

Email has met tremendous popularity over the past few years. People are sending and receiving many messages per day, communicating with partners and friends, or exchanging files and information. Unfortunately, the phenomenon of email overload has grown over the past years becoming a personal headache for users and a financial issue for companies. In this section, we will discuss how Machine Learning can contribute to the solution of this problem [13].

3.1 Automatic Answering

Large companies usually maintain email centres (in conjunction with "call centres") with employees committed to answer incoming messages. Those messages usually come from company clients and partners and many times address the same problems and queries. Automatic email answering is an effort to build email centers or personalized software that will be able to analyse an incoming message and then propose or even send an applicable answer. Efforts towards this direction have been made recently [14, 15].

3.2 Automatic Mail Organization into Folders

The growth of email usage has forced users to find ways to organize archive and manage their emails more efficiently. Many of them are organizing incoming messages into separate folders. Folders can be topic-oriented like "work", "personal" and "funny", people-specific like "John" and "Mary" or group-of-people-specific like "colleagues", "family" and "friends". Some users are archiving their messages according to importance and thus maintain folders like "urgent", "for future reference", "spam" etc. To achieve this, many users create manually some so-called *rules* to classify their email.

What Machine Learning has to offer to this task is the automatic classification of incoming email by observing past and current classifications made by the user (e.g. analyzing already existing folders or taking a current classification as an example). Thus, the user does not need to create the rules by himself. Furthermore, machine learning algorithms are able to classify a message, taking under consideration its content by searching for specific keywords. This is usually achieved by combining statistical and linguistic techniques. It is extremely convenient for the user, since there are some concepts like "messages concerning my work" or "interesting messages" or "messages that I have to answer today" that cannot easily be described with a combination of keywords. Moreover, these concepts may change (e.g. the concept of "interesting message") from time to time. A Machine Learning algorithm can learn to classify new messages just by silently observing past examples and can follow drift of concepts by accepting user feedback.

A lot of research has been recorded in the field [16, 17] and lots of those ideas have been implemented into useful email tools [18, 19].

3.3 Email and Thread Summarization

There is a certain category of email users that receive hundreds of messages per day. Some of them are newsletters, others are business decision-making messages from colleagues, appointment arrangements etc. It would be extremely useful for them if they could avoid reading all of those messages and instead read only the most important and necessary parts and then decide if the messages demand immediate attention. From a summary, they could also find out if a newsletter for example is interesting for them or not and only then read the full text. Again, data mining techniques are explored in order to build trainable tools for summarization. [20].

3.4 Spam Filtering

The main goal of spam filtering is to identify and sort out unsolicited commercial mails (spam) from a user's email stream. Spam email has begun as small annoyance in the early days of email to become a major industry problem in the last five years. The large amount of spam not only causes bandwidth (and therefore financial) problems, but also takes up valuable time from email users who try to separate and delete many unsolicited messages every day. Moreover, many spam messages include pornographic content inappropriate for children. Many different machine learning classifiers have been tested in the bibliography including Naïve Bayes [21], Support Vector Machines [22], Stacking Classifiers [23] and some of them have proved to be particularly accurate.

3.6 Dynamic Feature Space and Incremental Feature Selection for Email Classification

Email classification problems are of special interest for the Machine Learning and Data Mining community, mainly because they introduce and combine a number of special difficulties. They deal with high dimensional, streaming, unstructured, and, in many occasions, concept drifting data. Another important peculiarity of email (and streaming text in general), not adequately discussed in the relative literature, is the fact that the feature space is initially unavailable. In this section we present a computationally undemanding method that tackles with this problem [24].

Our approach uses two components in conjunction: a) an incremental feature ranking method, and b) an incremental learning algorithm that can consider a subset of the features during prediction. Feature selection methods that are commonly used for text classification are filters that evaluate the predictive power of each feature and select the N best. Such methods evaluate each word based on cumulative statistics concerning the number of times that it appears in each different class of documents. This renders such methods inherently incremental: When a new labeled document arrives, the statistics are updated and the evaluation can be immediately calculated without the need of re-processing past data. These methods can also handle new words by including them in the vocabulary and initializing their statistics. Therefore the first component of our approach can be instantiated using a variety of such methods, including information gain, the x^2 statistic or mutual information.

The incremental re-evaluation and addition of words will inevitably result into certain words being promoted to / demoted from the top N words. This raises a problem that requires the second component of the proposed approach: a learning algorithm that is able to classify a new instance taking into account different features over time. This problem has not been considered before to the best of our knowledge. We call learning algorithms that can deal with it *feature-based*, because learning is based on the new subset of features, in the same way that in instance based algorithms, learning is based on the new instance. An inherently feature based algorithms is Naive Bayes (NB) where, each feature makes an independent contribution towards the prediction of a class. Therefore, it can be easily expanded in order to instantiate the second component of our approach. Specifically, when NB is used for the classification of a new instance, it should also be provided with an additional parameter denoting the subset of the selected features it will then only consider the calculated probabilities of this subset. Figure 1 presents algorithm Update for the incremental update of our approach.

```
input  : Document, DocClass, Classes, Vocabulary
output: Classifier, Vocabulary, WordStats, Evaluation
begin
    foreach Word ∈ Document do
        if Word ∉ Vocabulary then
            AddWord(Word, Vocabulary)
            foreach Class ∈ Classes do
                WordStats [Word][Class][1] ← 0
                WordStats [Word][Class][0] ← 0
    foreach Word ∈ Vocabulary do
        if Word ∈ Document then
            WordStats [Word][DocClass][1] ← WordStats [Word][DocClass][1] + 1
        else
            WordStats [Word][DocClass][0] ← WordStats [Word][DocClass][0] + 1
    foreach Word ∈ Vocabulary do
        Evaluation ← EvaluateFeature(Word, WordStats)
    Classifier ← UpdateClassifier(Document, DocClass)
end
```

Figure 1 Algorithm for the incremental update.

4. Focused Crawling Using Reinforcement Learning

The World Wide Web can be considered the greatest library of all kinds of information that exists. In contrast with other kind of libraries, web lacks indexing structure, which allows the users to access the desired information. In order to deal with this problem search engines have been developed that try to organize and index web pages into extensive catalogues. Search engines try to collect as many pages as possible in order to cover the majority of the thematic topics. This process requires the use of programs called crawlers.

The architecture of these programs is simple. Starting from a set of seed pages, they crawl the web following the hyperlinks included there. Some initial approaches used graph based techniques like depth first search and breadth first search. Due to the explosive growth of the web the crawling that utilizes the above implementations became time and resource consuming process. Moreover, user groups that are interest in specific topics came up, which demanded for accurate information about these topics. Search engines are not capable to satisfy these needs and led to the construction of domain specific search engines.

Domain specific search engines attempt to collect pages that are relevant to a particular domain and them to their index. In order to achieve this goal, they utilize focused crawlers, which are agents that utilize the graph of the Web to find pages or documents that belong to a particular topic. Additionally, focused crawlers are used for the construction of thematic web portals and their maintenance with new and updated information.

A variety of methodologies have been proposed for the aforementioned problem. In [25] the crawler is based on link criteria whereas in [26] the system exploits the knowledge about domains in order to construct topic hierarchies. Other approaches [27, 28, 29], [30, 31] adopt learning and evolutionary algorithms respectively. In our work we frame the problem of focused crawling using Reinforcement Learning to learn an optimal crawling strategy.

4.1 Reinforcement Learning

Reinforcement Learning (RL) addresses the problem of how an agent can learn a behaviour through trial-and-error interactions with a dynamic environment [32]. In an RL task the agent, at each time step, senses the environment's state, s_t in S, where S is the finite set of possible states, and selects an action at in $A(s_t)$ to execute, where $A(s_t)$ is the finite set of possible actions in state s_t. The agent receives a reward, r_{t+1} in R, and moves to a new state s_{t+1}. The objective of the agent is to maximize the cumulative reward received over time. More specifically, the agent selects actions that maximize the expected discounted return:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

where γ is the discount factor and expresses the importance of future rewards. A *policy* π specifies that in state s the probability of taking an action α is $\pi(s,\alpha)$. For any policy π the *action-value function,* $Q^{\pi}(s,a)$, can be defined as the expected discounted return for executing a in state s and thereafter following π. The optimal policy, π^*, is the one that maximizes the action-value $Q^{\pi}(s,a)$ for all state-action pairs. In order to learn the optimal policy, the agent learns the optimal action-value function, Q^* which is defined as the expected return of taking action α in state s and thereafter following the optimal policy π^*. The most widely used algorithm for finding the optimal policy is the Q-learning algorithm [33].

4.2 Methodology

To formulate the problem of focused crawling as an RL problem we must define the following components:

A set of states, *S*: The state is defined as the pages accessed by the crawler, as the perception of the environment arises mainly by the pages retrieved at any given time.

A set of actions, *A*: Actions are defined as the categories of links, among which the crawler has to choose when visiting a specific page.

A reward function, *r(s,a)*: The crawler receives reward *r* when it visits page, s:

$$r = \begin{cases} 1 & \text{if } s = \text{target page} \\ 0 & \text{otherwise} \end{cases}$$

In the training phase the crawler executes a number of episodes starting from a seed set of pages. The episode ends when the crawler finds a relevant page or reaches a number of predefined steps. When the crawler visits a page it classifies the hyperlinks that the page contains, based on the textual information of the page that they link to. The crawler chooses a category of links from where it will select a URL to follow. The crawler receives a reward and transits to the next page. In order to explore the state space we make use of *ε-greedy* action selection method, where an action *a* is selected according to the following rule:

$$a = \begin{cases} \text{a random action with probability } \varepsilon \\ \arg\max_{a'} Q(s,a') \text{ with probability } 1-\varepsilon \end{cases}$$

In the crawling phase the crawler starts with a set of seed pages and uses a queue where it maintains a list of unvisited hyperlinks. The hyperlinks from each page are extracted and classified. Each action is evaluated using the function that crawler learned in the training phase. The crawler selects the link with the highest relevance score and repeats the same procedure until it visits a certain number of pages or when the queue is empty.

In order to evaluate the proposed methodology, the sport of snowboarding has been chosen as the specific topic for contacting the experiments. A dataset was collected by a breadth-first search crawl starting from the Winter Sports web page of dmoz[1]. We compared our approach with the widely used Best First strategy where the best hyperlink according to a criterion is selected and the page that it links to is fetched. We also implemented a variant of our approach with a different definition of the actions.

The results have shown that the proposed method approaches the behavior of the Best First strategy and outperforms the variant of the reinforcement learning approach. The results encourages as to improve the proposed method and also identify the types of task for which a learning agent might be better than the best-first heuristic.

[1] http://dmoz.org/Sports/Winter_Sports/

5. Conclusions

Because of the special characteristics of the new kind of data that are nowadays available (e.g. biological data), the variety of new problems and the extremely high importance of machine learning research, a large number of critical issues is still open and demands active and collaborative research by the academia as well as the industry. Moreover, new technologies led to a constantly increasing number of new questions on new data. The scientific community demands and machine learning provides the opportunities for novel and improved methods for analyzing all these data.

In this paper, some emerging trends of machine learning have been presented. Although mining from Web and biological data are considered as significantly upcoming topics in the field, there are also other interesting topics that were not discussed in this paper. Some of them are applications on MRI data, astronomical data, robotics, video games, music data etc. Other recent trends in the research of machine learning include learning from spatial and visual data. Machine learning can contribute to a better understanding of human interpretation and recognition of real world scenes, as well as of improving the capability of artificial vision systems. The most used learning method in computer vision problems is supervised learning and also Reinforcement Learning in some robotic vision tasks.

References

1 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26-28, 1993, 207-216.

2 Agrawal R, Srikant R. Mining sequential patterns. In Proceedings of the 11th IEEE ICDE International Conference on Data Engineering, pages 3-14, 1995.

3 Ma Q, Wang J T L. Biological data mining using Bayesian neural networks: A case study. International Journal on Artificial Intelligence Tools, Special Issue on Biocomputing, 1999, 8(4), 433-451.

4 Hirsh H, Noordewier M. Using background knowledge to improve inductive learning of DNA sequences. Proceedings of the 10th IEEE Conference on Artificial Intelligence for Applications, 1994, 351-357.

5 Zien A, Ratsch G, Mika S, Scholkopf B, Lengauer T, Muller R-K. Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics, 2000, 16(9), 799-807.

6 Houle J L, Cadigan W, Henry S, Pinnamaneni A, Lundahl S. Database mining in the human genome initiative. Whitepaper, Bio-databases.com, Amita Corporation., March 10th 2004, Available: http://www.biodatabases.com/ whitepaper.html

7 Li J, Ng K-S, Wong L. Bioinformatics adventures in database research. Proceedings of the 9th International Conference on Database Theory, 2003, Siena, Italy, 31-46.

8 Zeng F, Yap C H R, Wong L. Using feature generation and feature selection for accurate prediction of translation initiation sites. Genome Informatics, 2002, 13, 192-200.

9 Kozak M. An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. Nucleic Acids Research, 1987, 15(20) 8125-8148.

10 Tzanis G, Berberidis C, Alexandridou A, Vlahavas I. Improving the accuracy of classifiers for the prediction of translation initiation sites in genomic sequences, In Proceedings of the 10th Panhellenic Conference on Informatics, Bozanis P, Houstis E N (Eds.), Springer-Verlag, LNCS 3746, 426-436, Volos, Greece, 11-13 November, 2005.

11 Tzanis G, Vlahavas I. Prediction of translation initiation sites using classifier selection. In Proceedings of the 4th Hellenic Conference on Artificial Intelligence, Antoniou G, Potamias G, Plexousakis D, Spyropoulos C (Eds.), Springer-Verlag, LNAI 3955, 367-377, Heraklion, Crete, 18-20 May, 2006.

12 Pedersen A G, Nielsen H. Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, California, USA (1997) 226-233.

13 Katakis I, Tsoumakas G, Vlahavas I. Email mining: Emerging techniques for email management (to appear), Web Data Management Practices: Emerging Techniques and Technologies, Athena Vakali, George Pallis (Ed.), Idea Group Publishing, 32, 2006.
14 Bickel S, Scheffer T. Learning from message pairs for automatic email answering. In Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy, 2004.
15 Busemann S, Schmeier S, Arens R G. Message classification in the call center In Proceedings of the 6th Conference on Applied Natural Language Processing, Seattle, Washington, 2000.
16 Clarck J, Koprinska I, Poon J. A neural network based approach to automated e-mail classification. In Proceedings of the IEEE/WIC International Conference on Web Intelligence, 2003.
17 Klimt B, Yang Y. The Enron corpus: A new dataset for email classification research. In Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy, 2004.
18 Graham-Cumming J. PopFile: Automatic email classification (http://popfile.sourceforge.net), 2002.
19 Ho V. EMMA: An e-mail management assistant. In Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), 13-17 October 2003, Halifax, Canada.
20 Muresan S, Tzoukerman E, Klavans, J L. Combining linguistic and machine learning techniques for email summarization. In Proceedings of CoNLL-2001, Toulouse, France, 2001.
21 Sahami M, Dumais S, Heckerman D, Horvitz E. A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 Workshop. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998.
22 Drucker H, Vapnik V, We D. Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 1999, 10(5), 1048-1054.
23 Sakkis G, Androutsopoulos I, Paliouras G, Karkaletsis V, Spyropoulos C D, Stamatopoulos P. A memory-based approach to anti-spam filtering for mailing lists. Information Retrieval, 2003, 6(1), 49-73.
24 Katakis I, Tsoumakas G, Vlahavas I. On the utility of incremental feature selection for the classification of textual data streams, In Proceedings of the 10th Panhellenic Conference on Informatics, Bozanis P, Houstis E N (Eds.), Springer-Verlag, LNCS 3746, 338-348, Volos, Greece, 11-13 November, 2005.
25 Cho J., Garcia-Molina H, Page L. Efficient crawling through URL ordering. Computer Networks and ISDN Systems, 1998, 30(1-7) 161-172.
26 Chakrabarti S, Van den Berg M, Dom B. Focused crawling: a new approach to topic specific Web resource discovery. Computer Networks, 1999, 31(11-16) 1623-1640.
27 O'Meara T, Patel A. A topic specific web robot model based on restless bandits. IEEE Internet Computing, 2001, 5(2) 27-35.
28 Rennie J, McCallum A. Efficient web spidering with reinforcement learning. In Proceedings of the 16th International Conference on Machine Learning, 1999, 335-343.
29 Grigoriadis A, Paliouras G. Focused crawling using temporal difference learning. In Proceedings of the 3rd Hellenic Conference on Artificial Intelligence, 2004, 142-153.
30 Johnson J, Tsioutsiouliklis K, Giles C L. Evolving strategies for focused web crawling. In Proceedings of the 20th International Conference on Machine Learning.
31 Pant G, Menczer F. MySpiders: evolve your own intelligent web crawlers. autonomous agents and multi-agent systems, 2002, 5(2) 221-229.
32 Sutton R S, Barto A G. Reinforcement Learning, An Introduction. MIT Press, 1999.
33 Watkins C, Dayan P. Q-learning. Machine Learning, 1992, 8, 279-292.