Classification of Web Documents Using a Graph-Based Model and Structural Patterns

Andrzej Dominik, Zbigniew Walczak, and Jacek Wojciechowski

Warsaw University of Technology, Institute of Radioelectronics
Nowowiejska 15/19, 00-665 Warsaw, Poland
A.Dominik@elka.pw.edu.pl, Z.Walczak@elka.pw.edu.pl,
J.Wojciechowski@ire.pw.edu.pl

Abstract. The problem of classifying web documents is studied in this paper. A graph-based instead of traditional vector-based model is used for document representation. A novel classification algorithm which uses two different types of structural patterns (subgraphs): contrast and common is proposed. This approach is strongly associated with the classical emerging patterns techniques known from decision tables. The presented method is evaluated on three different benchmark web documents collections for measuring classification accuracy. Results show that it can outperform other existing algorithms (based on vector, graph, and hybrid document representation) in terms of accuracy and document model complexity. Another advantage is that the introduced classifier has a simple, understandable structure and can be easily extended by the expert knowledge.

1 Introduction

Classification problems have been deeply researched due to variety of applications. They appear in different fields of science and industry and may be solved using different techniques, e.g. neural networks, rough sets. One important research domain is automated categorization of text and web documents based on their content. Development of fast and accurate algorithms is required by web and corporate search engines to provide better quality of service (quality of search results in terms of accuracy and speed) for their users.

Algorithms for document categorization and classification operate on different data representation. The most popular document representation is based on vectors. According to this model, each term in a document becomes a feature (dimension). The value of each dimension in a vector is the weight of appropriate term in a given document. Weight usually denotes the frequency of a particular term inside a document or some other measure based on frequency.

The vector model has numerous advantages. It is simple and can be used with traditional classification algorithms that operate on vectors containing numerical values (e.g. k-NN algorithm, artificial neural networks, decision trees). The most important disadvantage of vector model is that it concentrates only on words frequency and ignores other sources of information. Additional information may

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 67–78, 2007.

be obtained from both the structure of a text (e.g. the order in which the words appear or the location of a word within the document) and the structure of a document (e.g. markup elements (tags) inside HTML web document). Such information may be crucial and may greatly improve web documents classification accuracy.

Another document representation model is based on a graph. In this model terms refer to nodes. Nodes are connected by edges which provides both text and document structure information. This model overcomes major limitations of the vector model. On the other hand dealing with data represented by a graph is more complex than by vector (i.e. common graph operations such as graph isomorphism are NP-complete).

There are also mixed (hybrid) document models which use both representations: graph and vector. They were designed to overcome problems connected with simple representations. They capture structure information (using graph model) and represent relevant data using vector [13].

This paper makes a few important contributions. Firstly, the concepts of contrast and common subgraphs are extended and used for building a Contrast Common Patterns Classifier (CCPC). Secondly, some typical emerging patterns ideas are adapted to improve classification accuracy. Classification results for benchmark web document collections obtained by using the considered approach are provided and compared with existing algorithms using different document representations.

This paper is organized as follows. In Section 2 the state of the art in graph mining and document classification is briefly described. Graph based web document representation model is presented in Section 3. In Section 4 preliminary terminology on graph theory is introduced. The concept of our classifier and experiments results are described in Sections 5 and 6, respectively. Conclusions, final remarks and future work are in Section 7.

2 Related Work

In this section we review the state of the art in the areas associated with mining contrast and common graph patterns.

Contrast patterns are substructures that appear (appear frequently) in one class of objects and don't appear (appear infrequently) in other classes. In data mining patterns which uniquely identify certain class of objects are called jumping emerging patterns (JEP). Patterns common for different classes are called emerging patterns (EP). Concepts of jumping emerging patterns and emerging patterns have been deeply researched as a tool for classification purposes [7], [8], [11].

The concept of contrast subgraphs was studied in [18], [2]. Ting and Bailey [18] proposed an algorithm (containing backtracking tree and hypergraph traversal algorithm) for mining all disconnected contrast subgraphs from dataset.

Another relevant area to review is mining frequent structures. Frequent structure is a structure which appears in samples of a given dataset more frequently than the specified treshold. Agarwal and Srikant proposed an efficient algorithm

for mining frequent itemsets in the transaction database called Apriori. Similar algorithms were later proposed for mining frequent subgraphs from graphs dataset: [10], [12]. They were also used for the classification purposes [4].

Mining patterns in graphs dataset which fulfil given conditions is a much more challenging task than mining patterns in decision tables (relational databases). The most computationally complex tasks are isomorphism and automorphism. The first is proved to be NP-complete while the complexity of the other one is still not known. All the algorithms for solving the isomorphism problem present in the literature, have an exponential time complexity in the worst case but polynomial solution has not been yet disproved. A universal exhaustive algorithm for both of these problems was proposed in [19]. It operates on the matrix representation of graphs and tries to find a proper permutation of nodes. Search space can be greatly reduced by using nodes invariants and iterative partitioning [9]. Moreover multiple graph isomorphism problems can be efficiently performed with canonical labelling [15], [9]. Canonical label is a unique representation (code) of a graph such that two isomorphic graphs have the same canonical label.

Another important issue is generating all non-isomorphic subgraphs of a given graph. The algorithm for generating DFS (Depth First Search) code [20] can be used to enumerate all subgraphs and reduce the number of required isomorphism checking.

One of the most popular approaches for document classification is based on k-NN (k-Nearest Neighbors) method. Different similarity measures where proposed for different document representations. For vector representation the most popular is cosine measure [17], [16] while for graph representation distance based on maximum common graph is widely used [17], [16]. Recently methods based on hybrid document representations became very popular. They are reported to provide better results than methods using simple representations. Markov and Last [13] proposed an algorithm that uses hybrid representation. It extracts subgraphs from a graph that represents document then creates vector with boolean values indicating relevant subgraphs.

3 Graph Representation of Web Documents

In this section we present basic information on graph based models for web document representation.

There are numerous methods for creating graphs from documents. In [16] six major algorithms were described: standard, simple, n-distance, n-simple distance, absolute frequency and relative frequency. All of these methods use adjacency of terms. Some of these methods were specially designed to deal with web documents by including markup elements information. In our case we used standard document representation (previously reported as being the most effective) with slight modifications (simplifications). We refer to this representation as standard simplified. This method produces a labeled (both nodes and edges) undirected multigraph. Detailed information on creating graph according to standard simplified model is provided below.

Before converting web document (HTML document) into a graph some preprocessing steps are taken. Firstly all words that do not provide any meaningful information about document's domain (e.g. "the", "and", "of") are removed from the text. Subsequently simple steaming method is performed in order to determine those word forms which should be considered to be identical (e.g. "graph" and "graphs"). Lastly, frequency of words appearing in document is calculated and document is divided into three major sections: title, which contains the text related to the documents title and any provided keywords; link, which is text appearing in clickable hyperlinks on the document; and text, which comprises any of the readable text in the document (this includes link text but not title and keyword text).

The final graph model is created on the previously preprocessed document. It contains only one parameter - N which refers to the number of nodes in a graph representing given document. This parameter is responsible for reducing computational complexity. N most frequently appearing terms in the text are extracted. Each such unique word becomes a node labeled with the term it represents. Note that there is only a single node for each word even if a word appears more than once in a text. If word a and word b are adjacent somewhere in a document section s, then there is an undirected edge from the node corresponding to a to the node corresponding to b with an edge label s. An edge is not added to the graph if the words are separated by certain punctuation marks (such as a period, comma).

4 Preliminary Terminology

In this section we introduce some basic concepts and definitions [18], [5], [6], [3] that are used in the subsequent sections.

Graphs are assumed to be undirected, connected (any two vertices are linked by a path), labelled (both vertices and edges posses labels) multigraphs (parallel edges and loops are allowed). By the size of a graph we mean the number of its edges. Capital letters (G, S, ...) denote single graphs while calligraphic letters ($\mathcal{G}$, $\mathcal{N}$, $\mathcal{P}$, ...) denote sets of graphs.

Definition 1. *Labelled graph G is a quadruple (V, E, α, β), where V is a non-empty finite set of vertices, E is non-empty finite multiset of edges ($E \subseteq V \cup [V]^2$) and α, β are functions assigning labels to vertices and edges, respectively.*

Definition 2. *A graph $S = (W, F, \alpha, \beta)$ is a subgraph of $G = (V, E, \alpha, \beta)$ (written as $S \subseteq G$) if: (1) $W \subseteq V$ and(2) $F \subseteq E \cap (W \cup [W]^2)$.*

Definition 3. *Let $\mathcal{G}$ be a set of graphs and let $G' = (V', E', \alpha', \beta')$ and $G = (V, E, \alpha, \beta)$. We say that G' is isomorphic to G (written as $G' \simeq G$) if there is an injective function $f : V' \longrightarrow V$ such that: (1) $\forall e' = (u', v') \in E' \quad \exists e = (f(u'), f(v')) \in E$, (2) $\forall u' \in V', \alpha'(u') = \alpha(f(u'))$ and (3) $\forall e' \in E', \beta'(e') = \beta(f(e'))$. If $f : V' \longrightarrow V$ is a bijective function then G' is automorphic to G (written as $G' = G$). If G' is not isomorphic to G then we write $G' \not\simeq G$.*

A graph G' is $\mathcal{G}$-isomorphic (written as $G' \simeq \mathcal{G}$) if: (1) $\exists G \in \mathcal{G} : G' \simeq G$. A graph G' is not $\mathcal{G}$-isomorphic (written as $G' \not\simeq \mathcal{G}$) if: (1) $\forall G \in \mathcal{G} : G' \not\simeq G$.

Definition 4. *Given the set of graphs $\mathcal{G}_1, ..., \mathcal{G}_n$ and a graph $M_{\mathcal{G}_1,...,\mathcal{G}_n}$. $M_{\mathcal{G}_1,...,\mathcal{G}_n}$ is a common subgraph for $\mathcal{G}_1, ..., \mathcal{G}_n$ if: $\forall i \in \langle 1, n\rangle : M_{\mathcal{G}_1,...,\mathcal{G}_n} \simeq \mathcal{G}_i$. Set of all common subgraphs for $\mathcal{G}_1, ..., \mathcal{G}_n$ will be denoted by $\mathcal{M}_{\mathcal{G}_1,...,\mathcal{G}_n}$. Set of all minimal (with respect to size i.e. containing only one edge and either one or two vertices) common subgraphs for $\mathcal{G}_1, ..., \mathcal{G}_n$ will be denoted as $\mathcal{M}^{\text{Min}}_{\mathcal{G}_1,...,\mathcal{G}_n}$.*

Definition 5. *Given the sets of graphs $\mathcal{N}$ and a graph P. A graph $C_{P\to\mathcal{N}}$ is a contrast subgraph of P with respect to $\mathcal{N}$ if: (1) $C_{P\to\mathcal{N}} \simeq P$ and (2) $C_{P\to\mathcal{N}} \not\simeq \mathcal{N}$. It is minimal (with respect to isomorphism) if all of $C_{\mathcal{P}\to N}$'s strict subgraphs are not contrast subgraphs. Set of all minimal contrast subgraphs of P with respect to $\mathcal{N}$ will be denoted as $\mathcal{C}^{\text{Min}}_{P\to\mathcal{N}}$.*

Definition 6. *Given the sets of graphs $\mathcal{P} = \{P_1, ..., P_n\}$ and $\mathcal{N}$. Let $\mathcal{C}^{\text{Min}}_{Pi\to\mathcal{N}}$ be the set of all minimal contrast subgraphs of P_i with respect to $\mathcal{N}$, $i \in \langle 1, n\rangle$. $\mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}$ is a set of all minimal contrast subgraphs of $\mathcal{P}$ with respect to $\mathcal{N}$ if: (1) $\forall C \in \mathcal{C}^{\text{Min}}_{Pi\to\mathcal{N}}\ \ \exists J \in \mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}} : J \simeq C$, for $i \in \langle 1, n\rangle$, (2) $\forall J_1 \in \mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}\ \ \neg\exists J_2 \in \mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}} \setminus J_1 : J_2 \simeq J_1$.*

$\mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}$ contains all minimal subgraphs (patterns) which are in $\mathcal{P}$ (i.e. each subgraph in $\mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}$ is isomorphic to at least one graph from $\mathcal{P}$) and are not present in $\mathcal{N}$ (i.e. each subgraph in $\mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}$ is not isomorphic to any graph from $\mathcal{N}$). What is more $\mathcal{C}^{\text{Min}}_{\mathcal{P}\to\mathcal{N}}$ contains only minimal (with respect to size and isomorphism) subgraphs.

Definition 7. *Given the sets of graphs $\mathcal{G}$, $\mathcal{N}$, $\mathcal{P}$ and a graph G. Let $\mathcal{S} = \{G' \in \mathcal{G} : G \simeq G'\}$. Support of graph G in $\mathcal{G}$ is defined as follows:* $\text{supp}_{\mathcal{G}}(G) = \frac{\text{card}(\mathcal{S})}{\text{card}(\mathcal{G})}$, *where* $\text{card}(\mathcal{G})$ *denotes the cardinal number of set $\mathcal{G}$. Growth rate of G in favour of $\mathcal{P}$ against $\mathcal{N}$ is expressed as follows:* $\rho_{\mathcal{P}\to\mathcal{N}}(G) = \frac{\text{supp}_{\mathcal{P}}(G)}{\text{supp}_{\mathcal{N}}(G)}$.

5 Web Document Classification Algorithm

In this section we propose a classification algorithm called: CCPC (Contrast Common Patterns Classifier). We present only the general concept without implementation details.

The concept of contrast subgraph is directly associated with the concept of jumping emerging pattern (JEP). They both define a pattern (either subgraph or set of items) exclusive for one class of objects. Similarly, the common subgraphs are associated with emerging patterns (EP), i.e. patterns that are present in both classes of objects. Measures for classical emerging patterns designed for classification purposes are mainly based on the support of a pattern in different classes of objects. This section adapts some classical scoring schemes to be used with contrast and common subgraphs.

Let $\mathcal{G}$ be a set of training graphs (graphs used for the learning of a classifier) and G be a test graph (graph to be classified). Let $\mathcal{G}$ be divided into n disjoint

decision classes: $\mathcal{G}_1, ..., \mathcal{G}_n$; $\mathcal{G} = \bigcup_{i=1..n} \mathcal{G}_i$. Let $\mathcal{C}^{\text{Min}}_{\mathcal{G}_i \to (\mathcal{G} \setminus \mathcal{G}_i)}$ be the set of all minimal contrast subgraphs of $\mathcal{G}_i$ with respect to graph set $(\mathcal{G} \setminus \mathcal{G}_i)$, where $i \in \langle 1, n \rangle$. Let $\mathcal{M}^{\text{Min}}_{\mathcal{G}_1,...,\mathcal{G}_n}$ be the set of all minimal common subgraphs for $\mathcal{G}_1, ..., \mathcal{G}_n$.

Let us now define a few score routines used for classification. Score is obtained using contrast subgraphs according to the following equations ($i \in \langle 1, n \rangle$):

$$\text{scConA}_{\mathcal{G}_i}(G) = \sum_{K \in \mathcal{K}} \text{supp}_{\mathcal{G}_i}(K), \quad \mathcal{K} = \{K : K \in \mathcal{C}^{\text{Min}}_{\mathcal{G}_i \to (\mathcal{G} \setminus \mathcal{G}_i)} \wedge K \simeq G\} \tag{1}$$

$$\text{scConB}_{\mathcal{G}_i}(G) = \frac{1}{\lambda_{\mathcal{G}_i}} * \sum_{K \in \mathcal{K}} \text{supp}_{\mathcal{G}_i}(K), \quad \mathcal{K} = \{K : K \in \mathcal{C}^{\text{Min}}_{\mathcal{G}_i \to (\mathcal{G} \setminus \mathcal{G}_i)} \wedge K \simeq G\} \tag{2}$$

where $\lambda_{\mathcal{G}_i}$ is a scaling factor. Scaling factors are median values from statistics of the contrast scores (1) determined for each graph from classes: $\mathcal{G}_1, ..., \mathcal{G}_n$.

Score is also calculated using common subgraphs according to the following equations:

$$\text{scComA}_{\mathcal{G}_i}(G) = \sum_{K \in \mathcal{K}} \text{supp}_{\mathcal{G}_i}(K), \quad \mathcal{K} = \{K : K \in \mathcal{M}^{\text{Min}}_{\mathcal{G}_1,...,\mathcal{G}_n} \wedge K \simeq G\} \tag{3}$$

$$\text{scComB}_{\mathcal{G}_i}(G) = \sum_{K \in \mathcal{K}} \rho_{\mathcal{G}_i \to (\mathcal{G} \setminus \mathcal{G}_i)}(G), \quad \mathcal{K} = \{K : K \in \mathcal{M}^{\text{Min}}_{\mathcal{G}_1,...,\mathcal{G}_n} \wedge K \simeq G\} \tag{4}$$

In (3) score depends directly on the support of the subgraphs, whereas in (4) it depends on the growth rate of certain patterns.

Classifier train process looks as follows. First all minimal (with respect to size and inclusion (non-isomorphic)) contrast subgraphs characteristic for each class are discovered ($\mathcal{C}^{\text{Min}}_{\mathcal{G}_i \to (\mathcal{G} \setminus \mathcal{G}_i)}$, where $i \in \langle 1, n \rangle$). Then all minimal (with respect to size and inclusion (non-isomorphic)) common subgraphs for all classes are discovered ($\mathcal{M}^{\text{Min}}_{\mathcal{G}_1,...,\mathcal{G}_n}$). Subgraph discovery is performed using DFS (Depth First Search) code [20] generation method and all necessary automorphism and isomorphism checking are performed using canonical labelling, nodes invariants, and iterative partitioning methods [15], [9]. Additional speed up can be achieved by limiting the size of discovered subgraphs i.e. instead of discover all contrast subgraphs only some of them are discovered up to given size (number of edges). This limitation may influence classification accuracy.

Classification process looks as follows. First scores based on contrast subgraphs are calculated for each class. We can choose between presented scoring schemes: scConA - we calculate $\text{scConA}_{\mathcal{G}_i}(G)$ from eq. (1) for each decision class; scConB - we calculate $\text{scConB}_{\mathcal{G}_i}(G)$ from eq. (2) for each decision class.

Test example G is assigned to a class with a highest score. If two or more decision classes have the highest score then G remains unclassified and scores based on common subgraphs are then calculated. Again we can choose one of the two approaches:scComA - we calculate $\text{scComA}_{\mathcal{G}_i}(G)$ from eq. (3) for each decision class;scComB - we calculate $\text{scComB}_{\mathcal{G}_i}(G)$ from eq. (4) for each decision class.

Test sample is assigned to a class with a higher score. If two or more decision classes have the highest score then G remains unclassified.

6 Experiments and Results

In order to evaluate the performance of the proposed classifiers we performed several experiments on three different benchmark collections of web documents, called F-series, J-series and K-series. The data comes from [1]. Each collection contains HTML documents which were originally news pages hosted at Yahoo (www.yahoo.com). Each document in every collection has a category (class) associated to the content of the document.

The original F-series collection contains 98 documents. Each of them is assigned to one or more (maximum three) of 17 subcategories of four major category areas. Some of the documents have conflicts subcategories (i.e. belonging to different major categories). We decided to remove those documents and simplify the problem by reducing the number of classes to major categories. The same operation was performed in [17], [13], [16], [14]. Final F-series collection contains 93 documents and four classes. Each document is assigned to one class.

The J-series collection contains 185 documents assigned to 10 categories while the K-series consists of 2340 documents belonging to 20 categories. In both cases each document is assigned to exactly one category.

We created one more web document collection, called K^{7th}-series. According to [17], this collection was created by selecting every 7^{th} document from K-series. It contains 335 documents representing 19 classes.

Summary of the benchmark document collections is provided in Table 1.

Table 1. Detailed information on F-series, J-series, K-series, and K^{7th}-series document collections

Document collection	Number of		Percentage of documents in category		
	documents	categories	minimal	median	maximal
F-series	93	4	20.4	24.7	28.0
J-series	185	10	8.6	10.3	10.8
K-series	2340	20	0.4	3.0	21.1
K^{7th}-series	335	19	0.6	3.9	21.2

We basically concentrated our research on the following issues: performance of classifiers and influence of training and test data complexity. Performance of classifiers (ability to assign the correct class to a document) was evaluated using leave-one-out cross-validation procedure. Accuracy of a classifier is expressed as the percentage of correctly classified documents. By data complexity we mean number of nodes in graph (N) representing each document.

Figures 1, 2, and 3 show classification accuracy of our method using different scoring routines as a function of number of nodes in a graph representing each document for different document collections. Figure 4 shows percentage of documents classified by contrast graphs for different document collections using scConB scoring scheme.

Table 2. Comparison of classification accuracy

Doc. series	Document model	Algorithm description	Number of nodes	Classification accuracy
F	Vector	k-NN, cosine	NA	94.6
	Vector	k-NN, Jaccard	NA	94.6
	Graph	k-NN, MCS	30	96.8
	Hybrid	k-NN, Naive	100	95.7
	Hybrid	k-NN, Smart	100	95.7
	Graph	**CCPC**	**30/50/100**	**91.4/98.9/98.9**
J	Vector	k-NN, cosine	NA	74.6
	Vector	k-NN, Jaccard	NA	77.3
	Graph	k-NN, MCS	30/60	85.4/86.5
	Hybrid	k-NN, Naive	30	87.6
	Hybrid	k-NN, Smart	40	94.6
	Graph	**CCPC**	**30/45**	**86.5/91.4**
K	Vector	k-NN, cosine	NA	77.5
	Vector	k-NN, Jaccard	NA	80.4
	Graph	k-NN, MCS	40/100/150	78.2/84.6/85.7
	Hybrid	k-NN, Naive	100	86.0
	Hybrid	k-NN, Smart	120	86.3
	Hybrid	C4.5, Naive	100	78.0
	Hybrid	NBC, Smart	100	76.0
	Graph	**CCPC**	**40**	**86.3**
K^{7th}	Vector	k-NN, cosine	NA	67.5
	Graph	k-NN, MCS	30/70	65.3/77.0
	Graph	**CCPC**	**30/70/100**	**59.4/73.7/84.8**

For all data collections scoring routines based on contrast subgraphs (scConA, scConB) has dominant influence on accuracy. Scoring routines based on common subgraphs (scComA, scComB) have in most cases very little impact on final results. As far as scoring on contrast subgraphs are concerned obtained with the scConB scoring scheme are more accurate than those obtained with scConA.

For most collections (except for J-series) classification accuracy is increasing with increasing number of nodes in a graph.

Table 2 shows comparison of accuracy for different classifiers based on different document representation. We made a selection of best available results for the following methods: k-NN with the vector representation, cosine and Jaccard similarity measure [17], [16]; k-NN with the graph representation and maximum common subgraph similarity measure [17], [16]; k-NN with the hybrid representation, Manhattan similarity measure, naive and smart subgraph extraction [13], [14]; C4.5 with the hybrid representation, Manhattan similarity measure, naive and smart subgraph extraction [14].

Our classifier (CCPC) outperforms other methods for F-series (smallest document collection) and K-series (largest document collection). For K-series our method required only 40 nodes while the runner-up required 120 to provide similar accuracy. For J-series collection CCPC method provided second best

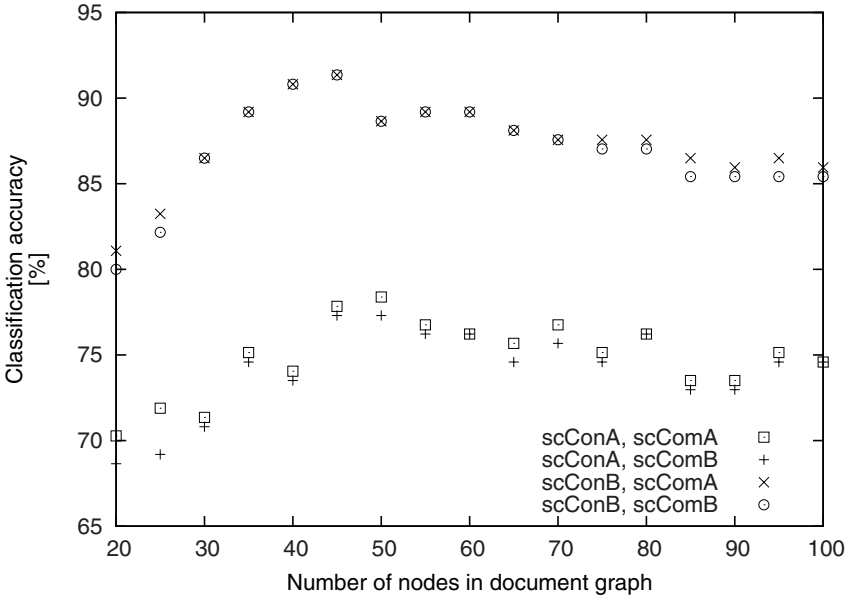

Fig. 1. Classification accuracy for J-series web document collection

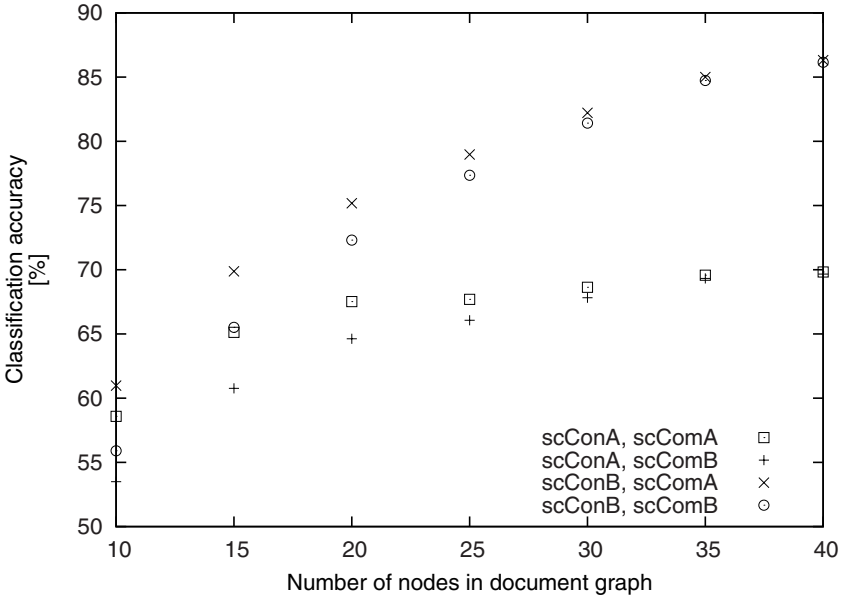

Fig. 2. Classification accuracy for K-series web document collection

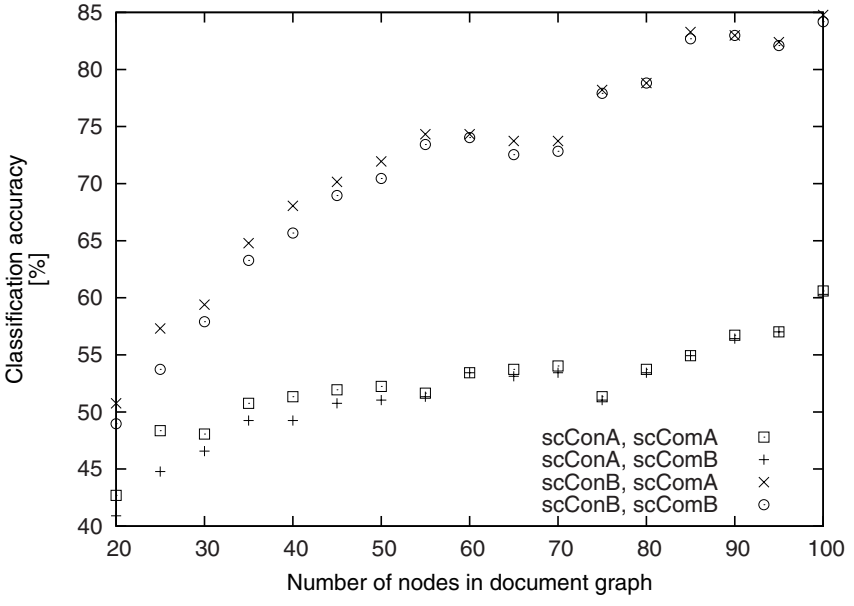

Fig. 3. Classification accuracy for K^{7th}-series web document collection

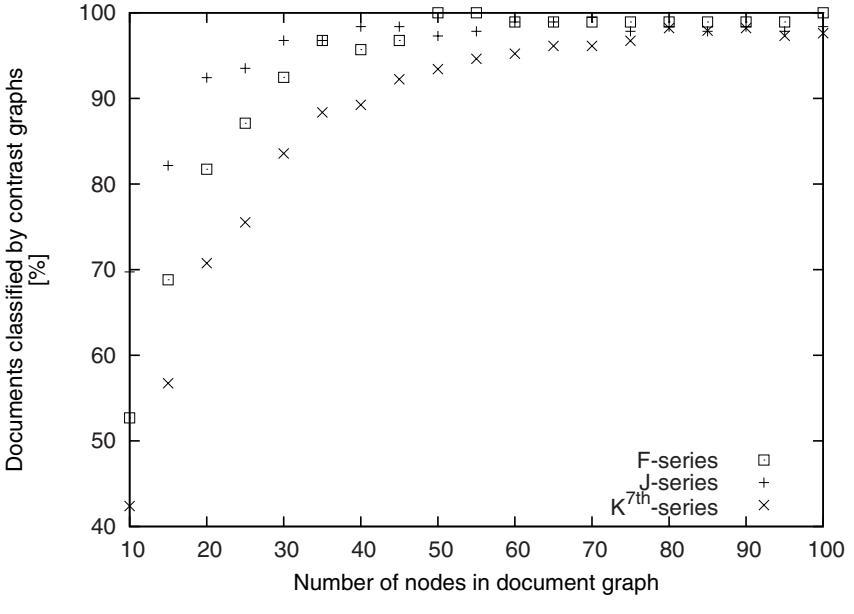

Fig. 4. Percentage of documents classified by contrast graphs for F-series, J-series and K^{7th}-series web document collections using scConB scoring scheme

result (after k-NN Smart). It is worth to mention that for all data collections we managed to achieve better results than methods based on vector document representation.

7 Conclusions

In this paper we presented a new approach for classifying web documents. Our algorithm (CCPC - Contrast Common Patterns Classifier) operates on a graph representation of a web document (HTML page) and uses concepts of contrast and common subgraphs as well as some ideas characteristic for emerging patterns technique.

The results show that our algorithm is competitive to existing schemes in terms of accuracy and data complexity (number of terms used for classification). For three document collections (F-series, K-series and K^{7th}) our method outperformed other approaches. What is more, construction and structure of our classifier is quite simple. This feature lets the domain expert to modify the classifier with professional knowledge by adding new patterns to contrast or common subgraphs sets or by modifying their supports.

The main concept of our classifier is domain independent so it can be used to solve classification problems in other areas (where data is represented by a graph) as well. It was already successfully applied in computational chemistry and chemical informatics for solving chemical compounds classification problems [6], i.e. detecting mutagenicity and carcinogenicity of chemical compounds for a given organism. The results shown that our algorithm outperformed the existing algorithms in terms of accuracy.

Our future research will concentrate on adapting the concept of contrast graphs to work with popular classification algorithm like k-NN. We will also try to apply our classifier in other research domains.

References

1. Datasets "pddpdata": `ftp://ftp.cs.umn.edu/dept/users/boley/pddpdata/`
2. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Washington, DC, USA, pp. 51–58. IEEE Computer Society Press, Los Alamitos (2002)
3. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters 19, 255–259 (1998)
4. Deshpande, M., Kuramochi, M., Karypis, G.: Frequent sub-structure-based approaches for classifying chemical compounds. In: ICDM '03: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'03), pp. 35–42 (2003)
5. Diestel, R.: Graph Theory. Springer, New York (2000)
6. Dominik, A., Walczak, Z., Wojciechowski, J.: Classifying chemical compounds using contrast and common patterns. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007, vol. 4432, pp. 772–781. Springer-Verlag, Berlin Heidelberg (2007)

7. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Knowledge Discovery and Data Mining, pp. 43–52 (1999)
8. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by aggregating emerging patterns. In: Discovery Science, pp. 30–42 (1999)
9. Fortin, S.: The graph isomorphism problem. Technical report, University of Alberta, Edomonton, Alberta, Canada (1996)
10. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Principles of Data Mining and Knowledge Discovery, pp. 13–23 (2000)
11. Kotagiri, R., Bailey, J.: Discovery of emerging patterns and their use in classification. In: Gedeon, T.D., Fung, L.C.C. (eds.) AI 2003. LNCS (LNAI), vol. 2903, pp. 1–12. Springer, Heidelberg (2003)
12. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01), pp. 313–320. IEEE Computer Society Press, Los Alamitos (2001)
13. Markov, A., Last, M.: Efficient graph-based representation of web documents. In: Proceedings of the Third International Workshop on Mining Graphs, Trees and Sequences (MGTS 2005), pp. 52–62 (2005)
14. Markov, A., Last, M., Kandel, A.: Model-based classification of web documents represenetd by graphs. In: Proceedings of WebKDD 2006: KDD Workshop on Web Mining and web Usage Analysis, iin conjunction with the 12th ACM SIGKDD International Conference on Knoowledge Discovery and Data Mining (KDD 2006), Philadelphia, PA, USA, ACM, New York (2006)
15. Read, R.C., Corneil, D.G.: The graph isomorph disease. Journal of Graph Theory 363, 339–363 (1977)
16. Schenker, A.: Graph-Theoretic Techniques for Web Content Mining. PhD thesis, University of South Florida (2003)
17. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of web documents using a graph model. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), vol. 01, pp. 240–244. IEEE Computer Society, Los Alamitos, CA, USA (2003)
18. Ting, R.M.H., Bailey, J.: Mining minimal contrast subgraph patterns. In: SIAM '06: Proceedings of the 2006 SIAM Conference on Data Mining, Maryland, USA (2006)
19. Ullmann, J.R.: An algorithm for subgraph isomorphism. J. ACM 23(1), 31–42 (1976)
20. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning (2002)