Feature Selection for Collective Classification

Baris Senliol
Istanbul Technical University,
Computer Engineering Department
Maslak, Istanbul, Turkey
Email: senliol@itu.edu.tr

Atakan Aral
Istanbul Technical University,
Computer Engineering Department
Maslak, Istanbul, Turkey
Email: aralat@itu.edu.tr

Zehra Cataltepe
Istanbul Technical University,
Computer Engineering Department
Maslak, Istanbul, Turkey
Email: cataltepe@itu.edu.tr

***Abstract*—When in addition to node contents and labels, relations (links) between nodes and some unlabeled nodes are available, collective classification algorithms can be used. Collective classification algorithms, like ICA (Iterative Classification Algorithm), determine labels for the unlabeled nodes based on the contents and/or labels of the neighboring nodes. Feature selection algorithms have been shown to improve classification accuracy for traditional machine learning algorithms. In this paper, we use a recent and successful feature selection algorithm, mRMR (Minimum Redundancy Maximum Relevance, Ding and Peng, 2003), on content features. On two scientific paper citation data sets, Cora and Citeseer, when only content information is used, we ahow that the selected features may result in almost as good performance as all the features. When feature selection is performed both on content and link information, even better classification accuracies are obtained. Feature selection considerably reduces the training time for both content only and ICA algorithms.**

Key Words: *Feature Selection, Collective Classification, Iterative Classification Algorithm (ICA), Minimum Redundancy Maximum Relevance (mRMR), Logistic Regression, Cora, Citeseer.*

I. Introduction

Learning problems with network information [1], [2], where for each node its features and relations with other nodes are available, become more common in our lives. Examples include social, financial, communication, electrical, computer, semantic, ecological, chemical reaction, gene regulatory and spin networks. Classification of nodes or links in the network, discovery of links or nodes which are not yet observed or identification of essential nodes or links, are some of the research areas on networked data. Availability of vast amount of nodes or features, unreliability of some of the link information are some of the common problems of these kind of data. Collective classification algorithms [4] consist of a set of classification algorithms for networked data. In collective classification, the content and link information for both training and test data are available. First, based on the available training content, link and label information, models are trained. Then, those models are used to label the test data simultaneously and iteratively where each test sample is labeled based on its neighbors.

Especially for problems with too many, correlated or noisy features, feature selection methods [3] have been used in pattern recognition for a long time. Feature selection not only helps with the accuracy of the learned models but also with the learning and testing time and explanation of decisions arrived at. mRMR (Minimum Redundancy Maximum Relevance) [11] is a recent fast and accurate feature selection method. mRMR uses both the input and label information when selecting features, but it does not actually train models, therefore is much faster than the wrapper [3] type feature selection methods such as forward or backward feature selection.

Feature selection for networked data is one of the new and interesting research topics. In this paper, we use feature selection on both content and link features. Our experiments on two paper citation datasets show that feature selection results in almost as accurate classifiers as the full feature set, while the training time of the classifiers are tremendously reduced with feature selection.

A. Collective Classification

In traditional machine learning, the observed and unobserved samples are assumed to be drawn independently from an identical distribution. Classification problems are solved using only samples' features (content) and labels. Connections/dependencies/relations between samples are not taken into consideration. However, in addition to content, connectivity information is often available and connectivity can be an important factor in determining the node labels. For example, papers which are on a certain topic, usually cite or are cited by other papers on the same topic, people who are interested in a certain product have close friends who could be

interested in the same product. Link-based classification takes into consideration the links between the objects in order to improve the estimation performance. Attributes of objects and links together can be considered as the features of the nodes. However, when two linked samples are not yet classified, they require each others labels to decide their own label. This situation could get even more complicated when links create cycles in a vast network [2]. Collective classification algorithms are proposed to overcome such problems.

Collective classification methods classify test objects in a network simultaneously. In these methods, the label of a sample in a network may affect the label of its neighbours, or even its neighbours' neighbours [1].

There exist collective classification algorithms that can infer exactly the right labels in a graph under certain conditions, but they are impractical and infeasable especially for large datasets. As a result, instead of exact inference which is thought to be an NP-Hard problem, approximate inference algorithms are developed [4]. Although, these algorithms are not always guaranteed to provide the right solution, they are feasible in terms of their computation time requirements.

Collective classification uses three types of information about a node, to determine the node's label in networked data:

1) The node's own observed attributes
2) Observed attributes and labels of its neighbours
3) Unobserved labels of its neighbours

The third type of information is required because all the nodes in a network are usually initially unlabeled [4]. In such situations, it is also required to bootstrapping is used to decide on the initial labels. One bootstraping method is to determine the initial labels by using only the first type of information (Content only) [5].

All collective classification algorithms take advantage of a base classifier to classify the nodes. Classifiers use link (relational) and content features [5].

Loopy belief propagation (LBP), mean field relaxation labeling (MF) and iterative classication algorithm (ICA) are popular approximate inference algorithms used for collective classification [2]. In this paper, we report our experiments using the ICA (Iterative Classification Algorithm) as our collective classification algorithm, whose details are given in Section II-A.

B. Feature Selection

As in any other classification method, in collective classification too, some features in the dataset may be noisy and/or irrelevant to the label or some subset of features may be adequate for labeling causing other features to be redundant. In such cases, determining and using a specific subset of features could result in a faster and more accurate solution.

A similar redundacy in features can be observed in the connections between the nodes, too. Some of the relations may worsen the performance of the classification, thus eliminating the unnecessary connections from the graph is beneficial in terms of both running time and accuracy.

Feature selection methods are divided into two categories according to their working principles: Feature selection/ordering methods create a group of features of desired number while feature subset selection methods create the best subset of features without intervention [7]. Another categorization of feature selection methods can be made according to how they interact with learning algorithms. Filter methods that are fast, scalable and modular, work independently from the learning algorithm. Wrapper methods, on the other hand, are guided for their search by a learning algorithm. They are less scalable and more likely to overfit data. They require training and validation, so they are slower, however they achieve better accuracy. Finally, embedded feature selected methods, that work faster than wrappers, are optimized specifically for a learning algorithm and they are installed into it [3]. All feature selection methods have 6 main characteristic properties which distinguish them. These properties are the initial state of search, creating successors, search strategy, feature evaluation method used, including or not including the interdependence of features and halting criterion [8].

The feature selection method chosen in this paper is the Minimum Redundancy Maximum Relevance Feature Selection (mRMR) [11], which is a filter method that makes use of the features and labels at the same time. Please see Section II-B for more details.

II. Methods

Detailed descriptions of ICA (Iterative Classification Algorithm), which is applied for collective classification and mRMR (minimum Redundancy Maximum Relevance) method, which is applied for feature selection are given in this section.

A. Iterative Classification Algorithm

To determine the label of a node, ICA assumes that all of the neighbours' attributes and labels of that node are already known. Then, it calculates the most likely label with a local classifier which uses node content and

neighbours' labels. However, it is extremely rare to find a node with all of its neighbours labelled. So, ICA repeats the process iteratively until all of the assignments stabilize. One problem is all nodes do not have equal number of neighbours. This makes it hard to implement the local classifier which should take constant number of inputs. To overcome this difficulty, an aggregation operator such as count, mode or exists is used. For example, count operator returns the number of the occurrences of each label in the neighbours [4].

In this paper, logistic regression is chosen as the local classifier of the iterative classification algorithm, while count aggregation operator is used to represent the relational features [2]. The algorithm for ICA (based on [4]) is given below. In pseudo code, OA represents observed attributes(content) of the samples. Y represents the unlabeled data and y_i stands for temporary label assignment of sample Y_i. g is the result of local classifier which shows the probability of getting label y_i for sample Y_i. O is the random ordering of nodes in every step of the iteration.

Iterative Classification Algorithm (ICA)

for all $Y_i \in Y$ **do**
  Compute $g(y_i|v_{N(Y_i);w})$ only using $OA(Y_i|x_{N(Y_i)})$ $\forall y_i \in C$.
  Set $y_i \leftarrow \arg\max_y g(y|v_{N(Y_i)})$
end for
repeat
  Generate Ordering O over nodes Y
  for all $Y_i \in O$ **do**
    Compute $g(y_i|v_{N(Y_i);w})$ $\forall y_i \in C$.
    $y_i \leftarrow \arg\max_y g(y|v_{N(Y_i)})$
  end for
until labels are stabilized

As shown above, the Iterative Classification Algorithm (ICA) starts with a bootstrapping to assign initial and temporary labels to all nodes by using only the content features of the nodes. Then, it starts iterating and updating labels according to the both relational and content features [9].

B. Minimum Redundancy Maximum Relevance

Identification of the most relevant features to the labels of the nodes can improve the efficiency of the classification process. One way of selecting features is to include features with highest correlation to the label, which is called maximum-relevance selection. An improvement to this method is including the features that correlate highest to the label but are as unrelated to each other as possible. This selection is more powerful than maximum-relevance selection as it minimizes redundancy [10]. Redundancy not only causes worse running time but it also reduces the accuracy of the classification.

To determine both feature-label and feature-feature correlations, mutual information is used. Mutual information measures the nonlinear correlations between features and is useful for both discrete or continuous variables. Mutual information for two discrete variables, x and y, is computed by using their marginal probabilities, $p(x)$ and $p(y)$, and their joint probabilistic distribution, $p(x, y)$ as shown below [11].

$$I(x, y) = \sum_{i,j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

Including the features which have high mutual information with the labels and excluding the features that have high mutual information among themselves is prefered for better identification of the most characteristic features [11].

III. Experiments

A. Data Sets

In this section, we give details on the CoRA and CiteSeer scientific citation datasets, which are used in the experiments below.

1) CoRA: CoRA data set consists of information on 2708 Machine Learning papers. Every paper in CoRA cites or is cited by at least one other paper in the data set. There are 1433 unique words that are contained at least 10 times in these papers. There are also 7 classes assigned to the papers according to their topics. For each paper, whether or not it contains a specific word, which class it belongs to, which papers it cites and which papers it is cited by are known. Citation connections and paper features (class and included words) are contained in two seperate files [12]. Total number of connections between the papers is 5429 [13]. There are 4.01 links per paper [14].

2) CiteSeer: CiteSeer data set consists of information on 3312 scientific papers. Every paper in CiteSeer also cites or is cited by at least one other paper in the data set. There are 3703 unique words that are contained at least 10 times in these papers. There are 6 classes assigned to the papers according to their topics. Just as in the CoRA dataset, word, class and cites and cited by information are given in two separate files [15]. Total number of connections between the papers is 4732 [13]. There are 2.77 links per paper [14].

Table I shows the total number of features in the last line and certain percentages of features for both CoRA and CiteSeer data sets. Iterative classification algorithms are run with feature selection, using the number of features given for each data set.

TABLE I
SELECTED NUMBERS OF FEATURES FOR DIFFERENT PERCENTAGES

Percentage	CoRA	CiteSeer
1	14	37
5	72	185
10	143	370
20	287	741
35	502	1296
100	1443	3703

B. Experimental Setup

This section describes the sampling of the data for the experiments and the base-classifier used.

1) Sampling: Data sets are splited into training and validation sets. Two methods are used for creating these partitions: k-fold cross-validation and snowball sampling.

In k-fold cross-validation nodes are splited into k random partitions. k times, k-1 of those partitions are used as train data, while the remaining one is left for validation [4]. Validation is performed k times to ensure that each subsample is used for validation exactly once and also to have an errorbar on the validation accuracy. When the number of links per node is low, k-fold sampling generates nearly disconnected graphs. k has been chosen to be 5 in our experiments.

To overcome the issue of disconnected graphs in k-fold cross validation, snow ball sampling is used. In this sampling method, after selecting the first node randomly, continuously the neighbours of the selected node are added to the subset. As a result, the generated subset is interconnected and balanced. Selected subset is used as the test data while remaining nodes compose the train subset. Snowball sampling is repeated k times to obtain k training-validation pairs [4].

2) Classifier: Logistic regression is used as the local classifier during the experiments. Logistic regression which is a discriminative model, determines conditional class probabilities without modelling the marginal distribution of features [16]. Logistic regression classifier can also be thought of as a one layer multilayer perceptron with its own special training algorithm.

C. Experimental Results

In the experiments, content only classification (CO) and ICA algorithm are compared for different percentages of features. When content only classification is performed, only the content features and node labels are used together with a logistic regression classifier. For each of the feature percentages given in table I, a separate set of k=5 training/validation experiments are performed. As it can be seen from table II, accuracy values generally increase along with the number of features. On the other hand, high accuracy values, nearly as high as the full data set, are achieved using only a small set of features. It is possible to get ICA results with acceptable accuracy with only 1% of the features in a much shorter time. The errorbars on the accuracies given in Table II are around 0.05.

When snowball sampling is used, the test accuracies are slightly higher for CiteSeer data set which contains less links per node, while k-fold sampling resulted in better accuracies than snowball sampling for CoRA data set which has more links.

It is also obvious from the results that iterative classification algorithm performs far better than the content only classification in all the experiments. This proves the importance of using network neighborhood information along with the content information.

Table III shows the average duration of k=5-fold experiments for each dataset and algorithm as the number of features increases. As seen in the table, as the number of features increase, algorithms take longer to run. With ICA, when only 5% of features are used, it is possible to spend only 28% (Cora) or 15% (Citeseer) of the time used for the full feature set. In addition, the accuracies achieved are still comparable to that of the full feature set.

TABLE III
TIME(SECONDS)

	Cora		Citeseer	
Percentage	CO	ICA	CO	ICA
1	6.102	8.298	4.296	8.838
5	8.059	8.248	6.583	11.967
10	10.503	10.655	10.919	16.555
20	18.259	11.523	25.481	24.311
35	22.412	18.493	32.753	33.969
100	67.785	28.938	92.533	82.741

Figures 1 and 2 show the weights for each feature (content and neighbor class count) at the end of the logistic regression training for logistic regression classifiers

TABLE II
RESULTS

	CoRA				CiteSeer			
	k-fold		Snowball		k-fold		Snowball	
Percentage	CO	ICA	CO	ICA	CO	ICA	CO	ICA
1	0.6022	0.8689	0.3807	0.4044	0.6537	0.7591	0.6743	0.7565
5	0.6899	0.8719	0.5296	0.5474	0.7024	0.7811	0.7112	0.7843
10	0.7146	0.8801	0.5770	0.6200	0.7000	0.7738	0.7142	0.7879
20	0.7416	0.8734	0.5333	0.5719	0.7293	0.7726	0.7251	0.7789
35	0.7506	0.8764	0.5874	0.6222	0.7146	0.7841	0.7227	0.7837
100	0.7588	0.8779	0.5956	0.6230	0.7232	0.7744	0.7353	0.7867

for each class for Cora and Citeseer datasets repectively. The neighbor class counts are represented by the last 7 features for the Cora dataset and the last 6 features for the Citeseer dataset. As seen in the figures, the weights corresponding to the neighbor class counts are much larger than the rest of the features, hence these are more important features for classification.

IV. CONCLUSIONS

In this paper we have shown a successful implementation of feature selection for a collective classification algorithm, ICA. The results show that, feature selection using the mRMR feature selection results in almost as good features as the whole dataset in terms of their accuracy. On the other hand, due to the reduced dimensionality of inputs, training takes much less time.

ACKNOWLEDGEMENTS

Authors Senliol and Cataltepe are supported by Tubitak research project 109E162. Author Cataltepe is also supported by a DPT (State Planning Organization) project.

REFERENCES

[1] Macskassy, S. and Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research.

[2] Sen, P. and Getoor, L. (2007). Link-based Classification. University of Maryland Technical Report CS-TR-4858.

[3] Guyon, I. and Elisseef, A. (2003). An introduction to variable and feature selection. In J. Machine Learning Res. 3, pp. 1157-1182.

[4] Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. AI Magazine on 22 September 2008 (Special Issue on AI and Networks)

[5] McDowell, L. K., Gupta, K. M., and Aha, D. W. (2007). Case-Based Collective Classification. Proceedings of the Twentieth International FLAIRS Conference

[6] Senliol, B., Gulgezen, G., and Cataltepe, Z. (2008). Fast Correlation Based Filter (FCBF) with a Different Search Strategy. International Symposium on Computer and Information Sciences (ISCIS 2008).

[7] Liu, H. and Motoda, H. (2008). Computational methods of feature selection. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series.

[8] Senliol, B., Cataltepe, Z., and Oommen, B. J. (2009). Feature Selection Using Supervised Feature Clustering with a Novel Learning-Automaton. Submitted to European Conferance of Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2009).

[9] Sen, P. and Getoor, L. (2006). Empirical Comparison of Approximate Inference Algorithms for Networked Data. ICML workshop on Open Problems in Statistical Relational Learning (SRL).

[10] Peng, H. C., Long, F. and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp.1226-1238

[11] Ding, C. and Peng, H. (2005). Minimum Redundancy Feature Selection from Microarray Gene Expression Data. Journal of Bioinformatics and Computational Biology

[12] McCallum, A., Nigam, K., Rennie, J. and Seymore, K. (2000). Automating the construction of internet portals with machine learning. Information Retrieval Journal.

[13] Statistical relational learning group. University of Maryland. http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html

[14] McDowell, L. K., Gupta, K. M. and Aha, D. W. (2007). Cautious Inference in Collective Classification. Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-07).

[15] Giles, C. L., Bollacker, K. and Lawrence, S. (1998). Citeseer: An automatic citation indexing system. In ACM Digital Libraries.

[16] Popescul, A.; and Ungar, L.H. 2003. Structural Logistic Regression for Link Analysis. 2nd Workshop on Multi-Relational Data Mining (MRDM 2003).

Fig. 1. Weights at the end of logistic regression training of ICA for Cora dataset.

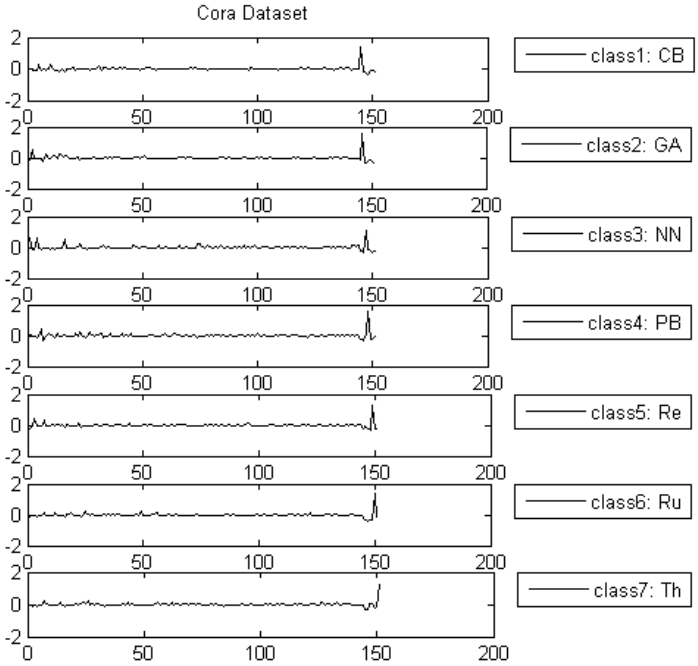

Fig. 2. Weights at the end of logistic regression training of ICA for Citeseer dataset.