
Collective Classification with Content and Link Noise

Baris Senliol*, Zehra Cataltepe, Abdullah Sonmez

Department of Computer Engineering

Istanbul Technical University

Maslak, Istanbul, Turkey

{senliol, cataltepe, sonmezab}@itu.edu.tr

Abstract

Collective classification algorithms [Mackassky et.al. 2007, Sen et.al. 2008] could be used for better classification of networked data when unlabeled test node features and links are available. In this study, we provide detailed results on the performance of collective classification algorithms when content or link noise is present. First of all, we show that collective classification algorithms are more robust to content noise than content only classification. We also evaluate the performance of collective classification when additive link noise is present. We show that, especially when content and/or link noise is present, feature and/or node selection is essential for better collective classification.

1 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 and gene regulatory 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 and unreliability of some of the link information are some of the common problems of these kinds of networks. Collective classification [2] is an approach for classifying unlabeled data which are in a network structure. 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 each test sample based on its neighbors, simultaneously and iteratively.

When inputs to a classification algorithm are noisy, in order to improve test classification performance, noise has been handled implicitly or explicitly. A particular implicit noise treatment approach is the pruning of decision trees as in [3]. [4] showed that explicit treatment of noise results in the same average performance but with less variance. Feature selection [5] has also been shown to improve classification accuracy when features are noisy. In order to deal with noisy links, [6] introduced the latent linkage semantic kernels. They used eigen value decomposition in kernel-induced space to filter out the unimportant links.

In this paper, first of all, we evaluate the performance of content only and collective classification, for different levels of content noise and on two datasets Cora and WebKB, whose details are given below. Then we show that using feature and node selection can improve the accuracy of classification accuracy. We show that when content and link noise is present, using feature and node selection results in great accuracy improvement.

The rest of the paper is organized as follows: In Section 2 we give the setup of the collective classification and also details on the collective classification algorithm that we use in the paper:

*Contact Author: senliol@itu.edu.tr.

Iterative Classification Algorithm (ICA). In Section 3 we give details on the content and link noise that we use in the paper. Section 4 summarizes the feature and node selection algorithms used in the paper. Section 5 gives detailed information about the datasets used in the experiments and the experimental methodology followed. Section 6 presents the experimental results. Section 7 concludes the paper.

2 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 other's labels to decide on 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.

2.1 Iterative Classification Algorithm

Iterative classification algorithm (ICA) is a successful approximate inference algorithm used for collective classification. Despite its simplicity, it performs sufficiently well or better than other collective classification algorithms, such as Gibbs Sampling [2].

To determine the label of a node, Iterative Classification Algorithm (ICA) assumes that all of the neighbors' 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 neighbors' labels. However, it is extremely rare to find a node with all of its neighbors labeled. So, ICA repeats the process iteratively until all of the label assignments become stabilized. One problem is all nodes do not have equal number of neighbors. 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 can be used. For example, count operator returns the number of occurrences of each label among the neighbors [2]. In this paper, logistic regression is chosen as the local classifier of the ICA, while count aggregation operator is used to represent the relational features [2]. Please see [2] for details of the ICA algorithm.

3 Content and Link Noise Generation

We added noise to content, to links and to both content and links. Content noise is added by means of flipping each feature of each node with certain probability. Flip probabilities of 0.1 and 0.3 are used in the experiments. Link noise is produced by means of adding links between instances with different labels as in [6]. The ratio of the number of noisy links to the number of original links was selected to be 0.1 or 0.2.

4 Feature and Node 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 which causes the other features to become redundant. In such cases, determining and using a specific subset of features results in a faster and usually more accurate solution [5].

A similar redundancy 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.

The feature selection method chosen in this paper is the Minimum Redundancy Maximum Relevance Feature Selection (mRMR) [7]. mRMR is a filter type feature selection method that makes use of the features and labels at the same time. mRMR chooses features which are relevant to the labels (maximum relevance). It also tries to come up with a feature subset that contains as non-redundant features as possible (minimum redundancy). To determine both feature-label and feature-feature correlations, mRMR uses the mutual information. Mutual information is a nonlinear correlation measure and can be used for both discrete and continuous variables. In [8] mRMR was shown to perform better than other feature selection methods. mRMR algorithm selects the features which have high mutual information with the labels (relevance) and low mutual information among themselves (redundancy) in a forward manner. MIQ (Mutual Information Quotient) or MID (Mutual Information Difference) criteria are used in order to achieve both high relevance and low redundancy. Please see [7] for more details on mRMR.

Node selection allows selection of certain nodes (and therefore links) in a network to improve the performance of collective classification. In this work, we selected the important nodes and used their links to construct the network. Unselected nodes are still in the dataset but their links to other nodes are not present in the network. In order to do node selection three different methods were considered: degree based, connectivity matrix mRMR, and neighbor consistency based. Since neighbor consistency based node selection were shown to outperform the other two methods in previous work [9], results using neighbor consistency based node selection is given in this paper. Neighbor Consistency Based Node Selection method selects nodes based on the agreement between their labels and their neighbors' labels. We define Neighbor Consistency of a node as the number of neighbors with the same label as the base node divided by the node's degree. After computing the neighbor consistency for every node, for a certain threshold value, all the nodes with lower neighbor consistency are eliminated. Threshold values of 0.1, 0.3, 0.5, 0.7 and 0.9 are used in the experiments.

5 Experimental Setup

5.1 Datasets

We used the CoRA and WebKB scientific publication and web page datasets in the experiments. Both datasets are downloaded from the Statistical Relational Learning Group web site (<http://www.cs.umd.edu/projects/linqs/projects/lbc>) at the University of Maryland. Cora dataset consists of information on scientific papers. WebKB is a dataset which consists of web pages from four different universities that are linking each other. As features, the words that occur at least 10 times are used. For each paper or page, whether or not it contains a specific word, which class it belongs to, which papers/pages it cites and which papers/pages it is cited by are known. Table 1 shows the total number of features, nodes, links, average degree and the number of classes for each dataset.

5.2 Sampling

When the usual k-fold cross-validation training and test sets are obtained on networked data, especially when the number of links per node is low, k-fold sampling generates nearly disconnected graphs [2]. To overcome the issue of disconnected graphs in k-fold cross validation, snowball sampling is used. In this sampling method, after selecting the first node randomly, continuously the neighbors 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 training set. Snowball sampling is repeated k times to obtain k training-test pairs [2]. In our experiments, we used 5-fold snowball sampling to measure performance of our experimental setups. In Cora dataset, there are plenty number of small and large networks but there is one particular network which has the most of the nodes (above 2000 nodes) and thus can be distinguished from others. Since our aim is to measure the performance of collective classification in a network, we have chosen these networks as core networks and built our 50-50 percent train-test splits only in these networks. We started from a random point in these networks and applied snowball sampling while preserving the class distribution of the dataset for test split. Nodes in the outside networks are added to training set to finalize test-train split. In WebKB datasets there four major networks instead of one that can be distinguished. As a result we started from four different random points in the four networks and applied snowball sampling as described above.

5.3 Local Classifier

A local classification method which is trained on node features and local connectivity information is needed for collective classification. We report experiments using logistic regression classifier which determines the conditional class probabilities without modeling the marginal distribution of features [10]. It can also be thought of as a one layer multilayer perceptron with a special training algorithm. In this paper, the regularized logistic regression whose details are given in [11] is used.

6 Experimental Results

We perform a number of experiments whose results are given in this section. First of all, we measure accuracy of collective and content only (CO) classification for different levels of content noise (Section 6.1). Then, we perform feature selection using mRMR and we show that while both CO and ICA benefit from feature selection, ICA benefits a lot more (Section 6.2). We also evaluate CO and ICA, using feature and node selection, on data with content and link noise (Section 6.3) and without noise (Section 6.4).

6.1 CO and ICA on data with noisy features

Table 2 presents the average accuracies obtained for different levels of content noise and when CO or ICA is the classification method. The table shows that for the Cora dataset, which has a lot more connectivity than WebKB, ICA helps in general whether there is content noise or not. For the WebKB dataset, ICA is better than CO only when the content noise is present.

6.2 Feature Selection

In this set of experiments, we show the behavior of collective classification versus content only classification when features are noisy. In Figure 1 for clean features (noise=0) and features with 0.1 and 0.3 flip probability, the classification accuracy with mRMR feature selection is shown for content only classification with logistic regression (CO) and ICA and logistic regression (ICA). According to the classification accuracy obtained with the full set of features, when features are noisy, collective classification continues to perform better than content only classification. On the other hand, when feature selection is used, collective classification performs remarkably well than content only classification. The difference is especially notable for the Cora dataset.

The figure shows that ICA achieves high accuracy values, nearly as high as the full data set, using only a small set of features. It is possible to get ICA results with acceptable accuracy with only 1 percent of the features. Using only 5 or 10 percent of features, ICA achieves better or as good accuracy as using all the features for both datasets. The error bars on the accuracies given in figure 1 are around 0.01.

We also experimented with two other methods, SVM-RFE (SVM Recursive Feature Elimination) of [12] and FCBF (Fast Correlation Based Filter) feature subset selection method of [13] to reduce the number of content features. SVM-RFE was too slow and FCBF could not reach the mRMR accuracy. So we do not give those results here.

6.3 Feature and node selection

In this section we report the accuracy results for collective classification with ICA and LR when both feature and node selection are used. Tables 3 and 4 show the accuracies obtained when different percentages of features (columns) and different node consistency thresholds (rows) are used. For both of the datasets, using feature and/or node selection results in better accuracies. Especially for the Cora dataset, using only a small number of features and nodes better classification accuracies are obtained. For the Cora dataset 5% of the features and nodes with a neighbor consistency of 0.7 or more, which correspond to 2034 nodes (75% of the original number of nodes) results in an average accuracy of 0.872, as opposed to the average accuracy of 0.784 when using all features and nodes. For the WebKB dataset, using only 20% of the features and all the nodes, again better classification accuracy of 0.854 than using all the features and nodes (0.816) is obtained.

6.4 Feature and node selection with content and link noise

In this set of experiments, the feature and node selection together with collective classification algorithms are used on data which have both feature and link noise. The feature and link noise levels are chosen to be both 0.1 for the first part and 0.3 and 0.2 respectively for the second part.

When both content and link noise levels are both 0.1, using both feature and node selection results in the best accuracies of 0.841 (Cora) and 0.793 (WebKB), while only feature selection results in the best accuracies of 0.766 (Cora), 0.793 (WebKB) (Table 5). Therefore, node selection helps with Cora dataset while it does not affect the WebKB results. Without feature or node selection, the accuracies with feature noise level of 0.1 are much lower at 0.585 (Cora) and 0.762 (WebKB).

Experiments also show that, when only links have an addition noise level of 0.1, feature and node selection improves the classification accuracies to 0.828 (Cora) and 0.856 (WebKB), from 0.777 (Cora) and 0.842 (WebKB) which are computed without node or feature selection.

When both feature and addition link noise 0.1 is present, feature and node selection, again results in better performance for Cora dataset, while the WebKB dataset again improves with feature selection and does not benefit from node selection (Table 5). When feature and link noise levels of 0.3 and 0.2 respectively are used (Table 6), similar behaviors of feature and node selection are observed on both datasets.

To show the statistical significance of the accuracy improvement when feature selection and node selection are used, paired t-tests are performed. Table 7 shows the t-test results, initial and best accuracy values that are obtained with our models for addition link and content noise. According to the t-tests, the accuracies obtained after the feature or node selection methods are significantly better than the original accuracies, without feature and node selection.

7 Conclusions and Future Work

In this paper, we have analyzed the performance collective classification algorithm ICA (Iterative Classification Algorithm) used with logistic regression as the local classification method. We have shown that collective classification achieves slightly better accuracy than content only classification when content noise is present. We have also shown that feature selection dramatically improves the performance of collective classification. When link noise is present, collective classification performance is affected. We have shown that node selection can be used to reduce the effect of noise. We have also used both content and node selection to remedy the negative effects of content and link noise.

One of the future work directions is the analysis of collective classification algorithms with feature and node selection when other classifiers are used. In their work, [2] have shown that logistic regression is better than other classification algorithms for collective classification. However, [14] states that decision trees are more capable of dealing with noise than logistic regression. Therefore, decision trees could be a good candidate to perform those experiments. In addition to analyzing input and link noise, analyzing noise on labels is also a possible future work direction. Another research direction is the measurement how content and link noise actually are on specific data sets and determination of the suitable local classifier, collective classification algorithm and feature/node selection method for that case.

Acknowledgements

Authors Senliol and Cataltepe are supported by Tubitak (The Scientific and Technological Research Foundation of Turkey) research project EEEAG 109E162.

References

- [1] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," May 2007.
- [2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, 2008.

- [3] J. Quinlan, “Simplifying decision trees,” 1987.
- [4] V. Mookerjee, M. Mannino, , and R. Gilson, “Improving the performance stability of inductive expert systems under input noise,” 1995.
- [5] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [6] Y. Tian, T. Huang, and W. Gao, “Latent linkage semantic kernels for collective classification of link data,” *J. Intell. Inf. Syst.*, vol. 26, no. 3, pp. 269–301, 2006.
- [7] C. H. Q. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” *J. Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–206, 2005.
- [8] H. Peng, F. Long, and C. H. Q. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [9] B. Senliol, Z. Cataltepe, and A. Sonmez, “Feature and node selection for collective classification,” 2009.
- [10] A. Popescul and L. H. Ungar, “Statistical relational learning for link prediction,” in *IJCAI Workshop on Learning Statistical Models from Relational Data, 2003*, 2003.
- [11] Q. Lu and L. Getoor, “Link based classification,” 2007.
- [12] Guyon, Weston, Barnhill, and Vapnik, “Gene selection for cancer classification using support vector machines,” *MACHLEARN: Machine Learning*, vol. 46, 2002.
- [13] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *ICML* (T. Fawcett and N. Mishra, eds.), pp. 856–863, AAAI Press, 2003.
- [14] C. Perlich, F. Provost, and J. S. Simonoff, “Tree induction vs. logistic regression: A learning-curve analysis,” 2003.

Table 1: Information about the datasets used in the experiments.

DataSet	Num Features	Num Nodes	Num Links	Average Degree	Num Classes
Cora	1443	3312	5429	3.9	6
WebKB	1703	877	1608	3.2	6

Table 2: Content Only (CO) and ICA Accuracies With Different Levels of Content Noise

DataSet	CO/ICA	Noise=0	Noise=0.1	Noise=0.3
Cora	CO	0.677	0.302	0.216
	ICA	0.784	0.585	0.203
WebKB	CO	0.821	0.74	0.52
	ICA	0.816	0.762	0.55

Figure 1: CO and ICA Results with noisy content.

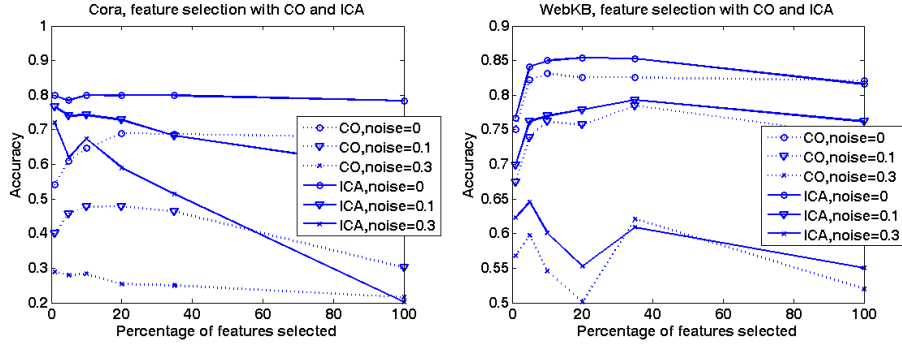


Table 3: Feature (with mRMR) and node selection (neighbor consistency based) accuracy (Cora dataset)

Node Consistency Threshold	Selected Percent of Features					No Feat. Sel.
	1	5	10	20	35	100
0.9	0.843	0.848	0.857	0.864	0.85	0.846
0.7	0.854	0.872	0.863	0.871	0.864	0.857
0.5	0.825	0.81	0.817	0.809	0.808	0.799
0.3	0.794	0.782	0.794	0.798	0.801	0.789
0.1	0.795	0.781	0.797	0.797	0.796	0.782
0.0 (No Node Sel.)	0.799	0.785	0.8	0.799	0.799	0.784

Table 4: Feature (with mRMR) and node selection (neighbor consistency based) accuracy (WebKB dataset)

Node Consistency Threshold	Selected Percent of Features					No Feat. Sel.
	1	5	10	20	35	100
0.9	0.753	0.821	0.831	0.828	0.825	0.813
0.7	0.754	0.821	0.83	0.827	0.825	0.813
0.5	0.754	0.821	0.832	0.828	0.825	0.814
0.3	0.753	0.822	0.831	0.829	0.827	0.816
0.1	0.759	0.832	0.846	0.844	0.841	0.819
0.0 (No Node Sel.)	0.767	0.841	0.85	0.854	0.853	0.816

Table 5: Feature and node selection accuracy, content noise=0.1, link noise=0.1

		Selected Percent of Features					No Feat. Sel.
	threshold	1	5	10	20	35	100
Cora	0.9	0.755	0.75	0.757	0.742	0.738	0.637
	0.7	0.771	0.753	0.77	0.763	0.704	0.679
	0.5	0.758	0.759	0.758	0.74	0.689	0.659
	0.3	0.743	0.74	0.735	0.718	0.664	0.587
	0.1	0.738	0.73	0.725	0.714	0.659	0.596
	0.0 (No Node Sel.)	0.737	0.734	0.725	0.71	0.667	0.583
WebKB	0.9	0.682	0.747	0.762	0.776	0.782	0.728
	0.7	0.682	0.748	0.764	0.778	0.783	0.729
	0.5	0.685	0.751	0.764	0.779	0.783	0.729
	0.3	0.676	0.745	0.763	0.781	0.787	0.721
	0.1	0.692	0.743	0.774	0.771	0.79	0.738
	0.0 (No Node Sel.)	0.706	0.769	0.786	0.791	0.803	0.759

Table 6: Feature and node selection accuracy, content noise=0.3, link noise=0.2

		Selected Percent of Features					No Feat. Sel.
	threshold	1	5	10	20	35	100
Cora	0.9	0.662	0.638	0.633	0.597	0.418	0.292
	0.7	0.668	0.644	0.63	0.569	0.407	0.272
	0.5	0.713	0.66	0.702	0.601	0.501	0.395
	0.3	0.723	0.704	0.699	0.584	0.419	0.31
	0.1	0.662	0.647	0.623	0.516	0.372	0.328
	0.0 (No Node Sel.)	0.663	0.637	0.635	0.523	0.358	0.335
WebKB	0.9	0.578	0.614	0.575	0.473	0.548	0.473
	0.7	0.565	0.609	0.548	0.465	0.54	0.482
	0.5	0.566	0.606	0.556	0.488	0.515	0.452
	0.3	0.573	0.604	0.553	0.493	0.514	0.506
	0.1	0.564	0.606	0.553	0.491	0.52	0.465
	0.0 (No Node Sel.)	0.642	0.676	0.683	0.635	0.669	0.552

Table 7: T-test results for FS and NS using ICA-LR on Noisy Content and Addition Link Noise

Dataset	Noise(Co-Li)	Init. Acc.	Best Acc.	p-value	Confidence Interval
Cora	0.1-0.1	0.583	0.771	0.007	[0.085 – 0.292]
	0.3-0.2	0.335	0.723	7.2×10^{-4}	[0.283 – 0.521]
WebKB	0.1-0.1	0.759	0.803	0.027	[0.025 – 0.062]
	0.3-0.2	0.552	0.683	0.04	[0.007 – 0.171]