Collective Classification of Congressional Floor-Debate Transcripts

Clinton Burfoot, Steven Bird and **Timothy Baldwin**
Department of Computer Science and Software Engineering
University of Melbourne, VIC 3010, Australia
{cburfoot, sb, tim}@csse.unimelb.edu.au

Abstract

This paper explores approaches to sentiment classification of U.S. Congressional floor-debate transcripts. Collective classification techniques are used to take advantage of the informal citation structure present in the debates. We use a range of methods based on local and global formulations and introduce novel approaches for incorporating the outputs of machine learners into collective classification algorithms. Our experimental evaluation shows that the mean-field algorithm obtains the best results for the task, significantly outperforming the benchmark technique.

1 Introduction

Supervised document classification is a well-studied task. Research has been performed across many document types with a variety of classification tasks. Examples are topic classification of newswire articles (Yang and Liu, 1999), sentiment classification of movie reviews (Pang et al., 2002), and satire classification of news articles (Burfoot and Baldwin, 2009). This and other work has established the usefulness of document classifiers as stand-alone systems and as components of broader NLP systems.

This paper deals with methods relevant to supervised document classification in domains with *network* structures, where *collective classification* can yield better performance than approaches that consider documents in isolation. Simply put, a network structure is any set of relationships between documents that can be used to assist the document classification process. Web encyclopedias and scholarly publications are two examples of document domains where network structures have been used to assist classification (Gantner and Schmidt-Thieme, 2009; Cao and Gao, 2005).

The contribution of this research is in four parts: (1) we introduce an approach that gives better than state of the art performance for collective classification on the ConVote corpus of congressional debate transcripts (Thomas et al., 2006); (2) we provide a comparative overview of collective document classification techniques to assist researchers in choosing an algorithm for collective document classification tasks; (3) we demonstrate effective novel approaches for incorporating the outputs of SVM classifiers into collective classifiers; and (4) we demonstrate effective novel feature models for iterative local classification of debate transcript data.

In the next section (Section 2) we provide a formal definition of collective classification and describe the ConVote corpus that is the basis for our experimental evaluation. Subsequently, we describe and critique the established benchmark approach for congressional floor-debate transcript classification, before describing approaches based on three alternative collective classification algorithms (Section 3). We then present an experimental evaluation (Section 4). Finally, we describe related work (Section 5) and offer analysis and conclusions (Section 6).

2 Task Definition

2.1 Collective Classification

Given a network and an object o in the network, there are three types of correlations that can be used

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1506–1515,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

to infer a label for o: (1) the correlations between the label of o and its observed attributes; (2) the correlations between the label of o and the observed attributes and labels of nodes connected to o; and (3) the correlations between the label of o and the unobserved labels of objects connected to o (Sen et al., 2008).

Standard approaches to classification generally ignore any network information and only take into account the correlations in (1). Each object is classified as an individual instance with features derived from its observed attributes. Collective classification takes advantage of the network by using all three sources. Instances may have features derived from their source objects or from other objects. Classification proceeds in a joint fashion so that the label given to each instance takes into account the labels given to all of the other instances.

Formally, collective classification takes a graph, made up of nodes $\mathcal{V} = \{V_1, \ldots, V_n\}$ and edges E. The task is to label the nodes $V_i \in \mathcal{V}$ from a label set $\mathcal{L} = \{L_1, \ldots, L_q\}$, making use of the graph in the form of a neighborhood function $\mathcal{N} = \{N_1, \ldots, N_n\}$, where $N_i \subseteq \mathcal{V} \setminus \{V_i\}$.

2.2 The ConVote Corpus

ConVote, compiled by Thomas et al. (2006), is a corpus of U.S. congressional debate transcripts. It consists of 3,857 speeches organized into 53 debates on specific pieces of legislation. Each speech is tagged with the identity of the speaker and a “for” or “against” label derived from congressional voting records. In addition, places where one speaker cites another have been annotated, as shown in Figure 1.

We apply collective classification to ConVote debates by letting $\mathcal{V}$ refer to the individual speakers in a debate and populating $\mathcal{N}$ using the citation graph between speakers. We set $\mathcal{L} = \{y, n\}$, corresponding to “for” and “against” votes respectively. The text of each instance is the concatenation of the speeches by a speaker within a debate. This results in a corpus of 1,699 instances with a roughly even class distribution. Approximately 70% of these are *connected*, i.e. they are the source or target of one or more citations. The remainder are *isolated*.

3 Collective Classification Techniques

In this section we describe techniques for performing collective classification on the ConVote corpus. We differentiate between *dual-classifier* and *iterative-classifier* approaches.

Dual-classifier approach: This approach uses a collective classification algorithm that takes inputs from two classifiers: (1) a *content-only* classifier that determines the likelihood of a y or n label for an instance given its text content; and (2) a *citation* classifier that determines, based on citation information, whether a given pair of instances are “same class” or “different class”.

Let Ψ denote a set of functions representing the classification preferences produced by the content-only and citation classifiers:

- For each $V_i \in \mathcal{V}$, $\phi_i \in \Psi$ is a function ϕ_i: $\mathcal{L} \rightarrow \mathbb{R}^+ \cup \{0\}$.
- For each $(V_i, V_j) \in E$, $\psi_{ij} \in \Psi$ is a function ψ_{ij}: $\mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}^+ \cup \{0\}$.

Later in this section we will describe three collective classification algorithms capable of performing overall classification based on these inputs: (1) the minimum-cut approach, which is the benchmark for collective classification with ConVote, established by Thomas et al.; (2) loopy belief propagation; and (3) mean-field. We will show that these latter two techniques, which are both approximate solutions for Markov random fields, are superior to minimum-cut for the task.

Figure 2 gives a visual overview of the dual-classifier approach.

Iterative-classifier approach: This approach incorporates content-only and citation features into a single *local* classifier that works on the assumption that correct neighbor labels are already known. This approach represents a marked deviation from the dual-classifier approach and offers unique advantages. It is fully described in Section 3.4.

Figure 3 gives a visual overview of the iterative-classifier approach.

For a detailed introduction to collective classification see Sen et al. (2008).

Debate 006

Speaker 400378 [against]

Mr. Speaker, . . . all over Washington and in the country, people are talking today about the majority's last-minute decision to abandon . . .

. . .

Speaker 400115 [for]

. . .

Mr. Speaker, . . . I just want to say to the ***gentlewoman from New York*** *that every single member of this institution . . .*

. . .

Figure 1: Sample speech fragments from the ConVote corpus. The phrase *gentlewoman from New York* by speaker 400115 is annotated as a reference to speaker 400378.

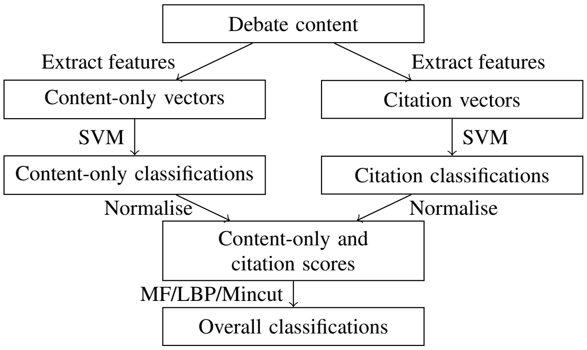

Figure 2: Dual-classifier approach.

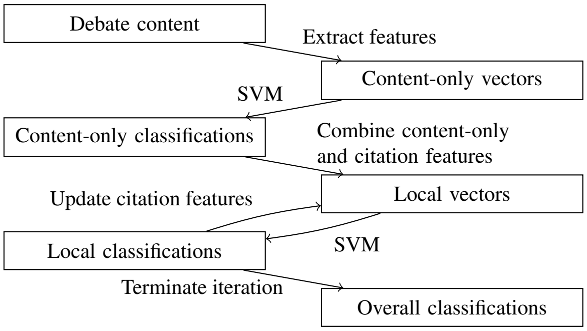

Figure 3: Iterative-classifier approach.

3.1 Dual-classifier Approach with Minimum-cut

Thomas et al. use linear kernel SVMs as their base classifiers. The content-only classifier is trained to predict y or n based on the unigram presence features found in speeches. The citation classifier is trained to predict "same class" or "different class" labels based on the unigram presence features found in the context windows (30 tokens before, 20 tokens after) surrounding citations for each pair of speakers in the debate.

The decision plane distance computed by the content-only SVM is normalized to a positive real number and stripped of outliers:

$$\phi_i(y) = \begin{cases} 1 & d_i > 2\sigma_i; \\ \left(1+\frac{d_i}{2\sigma_i}\right)/2 & |d_i| \leq 2\sigma_i; \\ 0 & d_i < -2\sigma_i \end{cases}$$

where σ_i is the standard deviation of the decision plane distance, d_i, over all of the instances in the debate and $\phi_i(n) = 1-\phi_i(y)$. The citation classifier output is processed similarly:[1]

$$\psi_{ij}(y,y) = \begin{cases} 0 & d_{ij} < \theta; \\ \alpha \cdot d_{ij}/4\sigma_{ij} & \theta \leq d_{ij} \leq 4\sigma_{ij}; \\ \alpha & d_{ij} > 4\sigma_{ij} \end{cases}$$

where σ_{ij} is the standard deviation of the decision plane distance, d_{ij} over all of the citations in the debate and $\psi_{ij}(n,n) = \psi_{ij}(y,y)$. The α and θ variables are free parameters.

A given class assignment v is assigned a cost that is the sum of per-instance and per-pair class costs derived from the content-only and citation classifiers respectively:

$$c(v) = \sum_{V_i \in \mathcal{V}} \phi_i(\bar{v}_i) + \sum_{(V_i,V_j)\in E: v_i \neq v_j} \psi_{ij}(v_i, v_i)$$

where v_i is the label of node V_i and $\bar{v}_i$ denotes the complement class of v_i.

[1] Thomas et al. classify each citation context window separately, so their ψ values are actually calculated in a slightly more complicated way. We adopted the present approach for conceptual simplicity and because it gave superior performance in preliminary experiments.

The cost function is modeled in a flow graph where extra source and sink nodes represent the y and n labels respectively. Each node in $\mathcal{V}$ is connected to the source and sink with capacities $\phi_i(y)$ and $\phi_i(n)$ respectively. Pairs classified in the "same class" class are linked with capacities defined by ψ.

An exact optimum and corresponding overall classification is efficiently computed by finding the minimum-cut of the flow graph (Blum and Chawla, 2001). The free parameters are tuned on a set of held-out data.

Thomas et al. demonstrate improvements over content-only classification, without attempting to show that the approach does better than any alternatives; the main appeal is the simplicity of the flow graph model. There are a number of theoretical limitations to the approach, which we now discuss.

As Thomas et al. point out, the model has no way of representing the "different class" output from the citation classifier and these citations must be discarded. This, to us, is the most significant problem with the model. Inspection of the corpus shows that approximately 80% of citations indicate agreement, meaning that for the present task the impact of discarding this information may not be large. However, the primary utility in collective approaches lies in their ability to fill in gaps in information not picked up by content-only classification. All available link information should be applied to this end, so we need models capable of accepting both positive and negative information.

The normalization techniques used for converting SVM outputs to graph weights are somewhat arbitrary. The use of standard deviations appears problematic as, intuitively, the strength of a classification should be independent of its variance. As a case in point, consider a set of instances in a debate all classified as similarly weak positives by the SVM. Use of ψ_i as defined above would lead to these being erroneously assigned the maximum score because of their low variance.

The minimum-cut approach places instances in either the positive or negative class depending on which side of the cut they fall on. This means that no measure of classification *confidence* is available. This extra information is useful at the very least to give a human user an idea of how much to trust the classification. A measure of classification confidence may also be necessary for incorporation into a broader system, e.g., a meta-classifier (Andreevskaia and Bergler, 2008; Li and Zong, 2008).

Tuning the α and θ parameters is likely to become a source of inaccuracy in cases where the tuning and test debates have dissimilar link structures. For example, if the tuning debates tend to have fewer, more accurate links the α parameter will be higher. This will not produce good results if the test debates have more frequent, less accurate links.

3.2 Heuristics for Improving Minimum-cut

Bansal et al. (2008) offer preliminary work describing additions to the Thomas et al. minimum-cut approach to incorporate "different class" citation classifications. They use post hoc adjustments of graph capacities based on simple heuristics. Two of the three approaches they trial appear to offer performance improvements:

The *SetTo* heuristic: This heuristic works through E in order and tries to force V_i and V_j into different classes for every "different class" ($d_{ij} < 0$) citation classifier output where $i < j$. It does this by altering the four relevant content-only preferences, $\phi_i(y)$, $\phi_i(n)$, $\phi_j(y)$, and $\phi_j(n)$. Assume without loss of generality that the largest of these values is $\phi_i(y)$. If this preference is respected, it follows that V_j should be put into class n. Bansal et al. instantiate this chain of reasoning by setting:

- $\phi'_i(y) = \max(\beta, \phi_i(y))$
- $\phi'_j(n) = \max(\beta, \phi_j(n))$

where ϕ' is the replacement content-only function, β is a free parameter $\in (.5, 1]$, $\phi'_i(n) = 1 - \phi'_i(y)$, and $\phi'_j(y) = 1 - \phi'_j(y)$.

The *IncBy* heuristic: This heuristic is a more conservative version of the *SetTo* heuristic. Instead of replacing the content-only preferences with fixed constants, it increments and decrements the previous values so they are somewhat preserved:

- $\phi'_i(y) = \min(1, \phi_i(y) + \beta)$
- $\phi'_j(n) = \min(1, \phi_j(n) + \beta)$

There are theoretical shortcomings with these approaches. The most obvious problem is the arbitrary nature of the manipulations, which produce a flow graph that has an indistinct relationship to the outputs of the two classifiers.

Bensal et al. trial a range of β values, with varying impacts on performance. No attempt is made to demonstrate a method for choosing a good β value. It is not clear that the tuning approach used to set α and θ would be successful here. In any case, having a third parameter to tune would make the process more time-consuming and increase the risks of incorrect tuning, described above.

As Bansal et al. point out, proceeding through E in order means that earlier changes may be undone for speakers who have multiple "different class" citations.

Finally, we note that the confidence of the citation classifier is not embodied in the graph structure. The most marginal "different class" citation, classified just on the negative side of the decision plane, is treated identically to the most confident one furthest from the decision plane.

3.3 Dual-classifier Approach with Markov Random Field Approximations

A pairwise Markov random field (Taskar et al., 2002) is given by the pair (G, Ψ), where G and Ψ are as previously defined, Ψ being re-termed as a set of *clique potentials*. Given an assignment v to the nodes $\mathcal{V}$, the pairwise Markov random field is associated with the probability distribution:

$$P(v) = \frac{1}{\mathcal{Z}} \prod_{V_i \in \mathcal{V}} \phi_i(v_i) \prod_{(V_i,V_j) \in E} \psi_{ij}(v_i, v_j)$$

where:

$$\mathcal{Z} = \sum_{v'} \prod_{V_i \in \mathcal{V}} \phi_i(v'_i) \prod_{(V_i,V_j) \in E} \psi_{ij}(v'_i, v'_j)$$

and v'_i denotes the label of V_i for an alternative assignment in v'.

In general, exact inference over a pairwise Markov random field is known to be NP-hard. There are certain conditions under which exact inference is tractable, but real-world data is not guaranteed to satisfy these. A class of approximate inference algorithms known as *variational methods* (Jordan et al., 1999) solve this problem by substituting a simpler "trial" distribution which is fitted to the Markov random field distribution.

Loopy Belief Propagation: Applied to a pairwise Markov random field, loopy belief propagation is a message passing algorithm that can be concisely expressed as the following set of equations:

$$\begin{aligned} m_{i \to j}(v_j) &= \alpha \sum_{v_i \in \mathcal{L}} \{\psi_{ij}(v_i, v_j) \phi_i(v_i) \\ &\qquad \prod_{V_k \in \mathcal{N}_i \cap \mathcal{V} \backslash V_j} m_{k \to i}(v_i), \forall v_j \in \mathcal{L}\} \\ b_i(v_i) &= \alpha \phi_i(v_i) \prod_{V_j \in \mathcal{N}_i \cap \mathcal{V}} m_{j \to i}(v_i), \forall v_i \in \mathcal{L} \end{aligned}$$

where $m_{i \to j}$ is a message sent by V_i to V_j and α is a normalization constant that ensures that each message and each set of marginal probabilities sum to 1. The algorithm proceeds by making each node communicate with its neighbors until the messages stabilize. The marginal probability is then derived by calculating $b_i(v_i)$.

Mean-Field: The basic mean-field algorithm can be described with the equation:

$$b_j(v_j) = \alpha \phi_j(v_j) \prod_{V_i \in \mathcal{N}_j \cap \mathcal{V}} \prod_{v_i \in \mathcal{L}} \psi_{ij}^{b_i(v_i)}(v_i, v_j), v_j \in \mathcal{L}$$

where α is a normalization constant that ensures $\sum_{v_j} b_j(v_j) = 1$. The algorithm computes the fixed point equation for every node and continues to do so until the marginal probabilities $b_j(v_j)$ stabilize.

Mean-field can be shown to be a variational method in the same way as loopy belief propagation, using a simpler trial distribution. For details see Sen et al. (2008).

Probabilistic SVM Normalisation: Unlike minimum-cut, the Markov random field approaches have inherent support for the "different class" output of the citation classifier. This allows us to apply a more principled SVM normalisation technique. Platt (1999) describes a technique for converting the output of an SVM classifier to a calibrated posterior probability. Platt finds that the posterior can be fit using a parametric form of a sigmoid:

$$P(y = 1|d) = \frac{1}{1 + \exp(Ad + B)}$$

This is equivalent to assuming that the output of the SVM is proportional to the log odds of a positive example. Experimental analysis shows error rate is improved over a plain linear SVM and probabilities are of comparable quality to those produced using a regularized likelihood kernel method.

By applying this technique to the base classifiers, we can produce new, simpler Ψ functions, $\phi_i(y) = P_i$ and $\psi_{ij}(y, y) = P_{ij}$ where P_i is the probabilistic normalized output of the content-only classifier and P_{ij} is the probabilistic normalized output of the citation classifier.

This approach addresses the problems with the Thomas et al. method where the use of standard deviations can produce skewed normalizations (see Section 3.1). By using probabilities we also open up the possibility of replacing the SVM classifiers with any other model than can be made to produce a probability. Note also that there are no parameters to tune.

3.4 Iterative Classifier Approach

The dual-classifier approaches described above represent *global* attempts to solve the collective classification problem. We can choose to narrow our focus to the *local* level, in which we aim to produce the best classification for a single instance with the assumption that all other parts of the problem (i.e. the correct labeling of the other instances) are solved.

The Iterative Classification Algorithm (Bilgic et al., 2007), defined in Algorithm 1, is a simple technique for performing collective classification using such a local classifier. After bootstrapping with a content-only classifier, it repeatedly generates new estimates for v_i based on its current knowledge of $\mathcal{N}_i$. The algorithm terminates when the predictions stabilize or a fixed number of iterations is completed. Each iteration is completed using a newly generated ordering $\mathcal{O}$, over the instances $\mathcal{V}$.

We propose three feature models for the local classifier.

Citation presence and Citation count: Given that the majority of citations represent the "same class" relationship (see Section 3.1), we can anticipate that content-only classification performance will be improved if we add features to represent the presence of neighbours of each class.

We define the function $c(i, l) = \sum_{v_j \in \mathcal{N}_i \cap \mathcal{V}} \delta_{v_j, l}$ giving the number of neighbors for node V_i with label l, where δ is the Kronecker delta. We incorporate these *citation count* values, one for the supporting class and one for the opposing class, obtaining a new feature vector $(u_i^1, u_i^2, \ldots, u_i^j, c(i, y), c(i, n))$ where $u_i^1, u_i^2, \ldots, u_i^j$ are the elements of $\vec{u}_i$, the binary unigram feature vector used by the content-only classifier to represent instance i.

Algorithm 1 Iterative Classification Algorithm

```
for each node $V_i \in \mathcal{V}$ do {bootstrapping}
    compute $\vec{a}_i$ using only local attributes of node
    $v_i \leftarrow f(\vec{a}_i)$
end for
repeat {iterative classification}
    randomly generate ordering $\mathcal{O}$ over nodes in $\mathcal{V}$
    for each node $V_i \in \mathcal{O}$ do
        {compute new estimate of $v_i$}
        compute $\vec{a}_i$ using current assignments to $\mathcal{N}_i$
        $v_i \leftarrow f(\vec{a}_i)$
    end for
until labels have stabilized or maximum iterations reached
```

Alternatively, we can represent neighbor labels using binary *citation presence* values where any non-zero count becomes a 1 in the feature vector.

Context window: We can adopt a more nuanced model for citation information if we incorporate the citation context window features into the feature vector. This is, in effect, a synthesis of the content-only and citation feature models. Context window features come from the product space $\mathcal{L} \times \mathcal{C}$, where $\mathcal{C}$ is the set of unigrams used in citation context windows and $\vec{c}_i$ denotes the context window features for instance i. The new feature vector becomes: $(u_i^1, u_i^2, \ldots, u_i^j, c_i^1, c_i^2, \ldots, c_i^k)$. This approach implements the intuition that speakers indicate their voting intentions by the words they use to refer to speakers whose vote is known. Because neighbor relations are bi-directional the reverse is also true: Speakers indicate other speakers' voting intentions by the words they use to refer to them.

As an example, consider the context window feature `AGREE-FOR`, indicating the presence of the *agree* unigram in the citation window *I agree with the gentleman from Louisiana*, where the label for the *gentleman from Louisiana* instance is y. This feature will be correctly correlated with the y label. Similarly, if the unigram were *disagree* the feature would be correlated with the n label.

4 Experiments

In this section we compare the performance of our dual-classifier and iterative-classifier approaches. We also evaluate the performance of the three feature models for local classification.

All accuracies are given as the percentages of instances correctly classified. Results are macro-averaged using 10 × 10-fold cross validation, i.e. 10 runs of 10-fold cross validation using different randomly assigned data splits.

Where quoted, statistical significance has been calculated using a two-tailed paired t-test measured over all 100 pairs with 10 degrees of freedom. See Bouckaert (2003) for an experimental justification for this approach.

Note that the results presented in this section are not directly comparable with those reported by Thomas et al. and Bansal et al. because their experiments do not use cross-validation. See Section 4.3 for further discussion of experimental configuration.

4.1 Local Classification

We evaluate three models for local classification: citation presence features, citation count features and context window features. In each case the SVM classifier is given feature vectors with both content-only and citation information, as described in Section 3.4.

Table 1 shows that context window performs the best with 89.66% accuracy, approximately 1.5% ahead of citation count and 3.5% ahead of citation presence. All three classifiers significantly improve on the content-only classifier.

These relative scores seem reasonable. Knowing the words used in citations of each class is better than knowing the number of citations in each class, and better still than only knowing which classes of citations exist.

These results represent an upper-bound for the performance of the iterative classifier, which relies on iteration to produce the reliable information about citations given here by oracle.

4.2 Collective Classification

Table 2 shows overall results for the three collective classification algorithms. The iterative classifier was run separately with citation count and context window citation features, the two best performing local classification methods, both with a threshold of 30 iterations.

Method	Accuracy (%)
Majority	52.46
Content-only	75.29
Citation presence	85.01
Citation count	88.18
Context window	89.66

Table 1: Local classifier accuracy. All three local classifiers are significant over the in-isolation classifier ($p < .001$).

Results are shown for connected instances, isolated instances, and all instances. Collective classification techniques can only have an impact on connected instances, so these figures are most important. The figures for all instances show the performance of the classifiers in our real-world task, where both connected and isolated instances need to be classified and the end-user may not distinguish between the two types.

Each of the four collective classifiers outperform the minimum-cut benchmark over connected instances, with the iterative classifier (context window) (79.05%) producing the smallest gain of less than 1% and mean-field doing best with a nearly 6% gain (84.13%). All show a statistically significant improvement over the content-only classifier. Mean-field shows a statistically significant improvement over minimum-cut.

The dual-classifier approaches based on loopy belief propagation and mean-field do better than the iterative-classifier approaches by an average of about 3%.

Iterative classification performs slightly better with citation count features than with context window features, despite the fact that the context window model performs better in the local classifier evaluation. We speculate that this may be due to citation count performing better when given incorrect neighbor labels. This is an aspect of local classifier performance we do not otherwise measure, so a clear conclusion is not possible. Given the closeness of the results it is also possible that natural statistical variation is the cause of the difference.

The performance of the minimum-cut method is not reliably enhanced by either the *SetTo* or *IncBy* heuristics. Only *IncBy*(.15) gives a very small improvement (0.14%) over plain minimum-cut. All of the other combinations tried diminished performance slightly.

4.3 A Note on Error Propagation and Experimental Configuration

Early in our experimental work we noticed that performance often varied greatly depending on the debates that were allocated to training, tuning and testing. This observation is supported by the per-fold scores that are the basis for the macro-average performance figures reported in Table 2, which tend to have large standard deviations. The absolute standard deviations over the 100 evaluations for the minimum-cut and mean-field methods were 11.19% and 8.94% respectively. These were significantly larger than the standard deviation for the content-only baseline, which was 7.34%. This leads us to conclude that the performance of collective classification methods is highly variable.

Bilgic and Getoor (2008) offer a possible explanation for this. They note that the cost of incorrectly classifying a given instance can be magnified in collective classification, because errors are propagated throughout the network. The extent to which this happens may depend on the random interaction between base classification accuracy and network structure. There is scope for further work to more fully explain this phenomenon.

From these statistical and theoretical factors we infer that more reliable conclusions can be drawn from collective classification experiments that use cross-validation instead of a single, fixed data split.

5 Related work

Somasundaran et al. (2009) use ICA to improve sentiment polarity classification of dialogue acts in a corpus of multi-party meeting transcripts. Link features are derived from annotations giving *frame* relations and *target* relations. Respectively, these relate dialogue acts based on the sentiment expressed and the object towards which the sentiment is expressed. Somasundaran et al. provides another argument for the usefulness of collective classification (specifically ICA), in this case as applied at a dialogue act level and relying on a complex system of annotations for link information.

Somasundaran and Wiebe (2009) propose an unsupervised method for classifying the stance of each contribution to an online debate concerning the merits of competing products. Concessions to other stances are modeled, but there are no overt citations in the data that could be used to induce the network structure required for collective classification.

Pang and Lee (2005) use metric labeling to perform multi-class collective classification of movie reviews. Metric labeling is a multi-class equivalent of the minimum-cut technique in which optimization is done over a cost function incorporating content-only and citation scores. Links are constructed between test instances and a set of k nearest neighbors drawn only from the training set. Restricting the links in this way means the optimization problem is simple. A similarity metric is used to find nearest neighbors.

The Pang and Lee method is an instance of implicit link construction, an approach which is beyond the scope of this paper but nevertheless an important area for future research. A similar technique is used in a variation on the Thomas et al. experiment where additional links between speeches are inferred via a similarity metric (Burfoot, 2008). In cases where both citation and similarity links are present, the overall link score is taken as the sum of the two scores. This seems counter-intuitive, given that the two links are unlikely to be independent. In the framework of this research, the approach would be to train a link meta-classifier to take scores from both link classifiers and output an overall link probability.

Within NLP, the use of LBP has not been restricted to document classification. Examples of other applications are dependency parsing (Smith and Eisner, 2008) and alignment (Cromires and Kurohashi, 2009). Conditional random fields (CRFs) are an approach based on Markov random fields that have been popular for segmenting and labeling sequence data (Lafferty et al., 2001). We rejected linear-chain CRFs as a candidate approach for our evaluation on the grounds that the arbitrarily connected graphs used in collective classification can not be fully represented in graphical format, i.e.

	Connected	Isolated	All
Majority	52.46	46.29	50.51
Content only	75.31	78.90	76.28
Minimum-cut	78.31	78.90	78.40
Minimum-cut (*SetTo*(.6))	78.22	78.90	78.32
Minimum-cut (*SetTo*(.8))	78.01	78.90	78.14
Minimum-cut (*SetTo*(1))	77.71	78.90	77.93
Minimum-cut (*IncBy*(.05))	78.14	78.90	78.25
Minimum-cut (*IncBy*(.15))	78.45	78.90	78.46
Minimum-cut (*IncBy*(.25))	78.02	78.90	78.15
Iterative-classifier (citation count)	80.07$\star$	78.90	79.69$\star$
Iterative-classifier (context window)	79.05	78.90	78.93
Loopy Belief Propagation	83.37†	78.90	81.93†
Mean-Field	84.12†	78.90	82.45†

Table 2: Speaker classification accuracies (%) over connected, isolated and all instances. The marked results are statistically significant over the content only benchmark ($\star\, p < .01$, † $p < .001$). The mean-field results are statistically significant over minimum-cut ($p < .05$).

linear-chain CRFs do not scale to the complexity of graphs used in this research.

6 Conclusions and future work

By applying alternative models, we have demonstrated the best recorded performance for collective classification of ConVote using bag-of-words features, beating the previous benchmark by nearly 6%. Moreover, each of the three alternative approaches trialed are theoretically superior to the minimum-cut approach approach for three main reasons: (1) they support multi-class classification; (2) they support negative and positive citations; (3) they require no parameter tuning.

The superior performance of the dual-classifier approach with loopy belief propagation and mean-field suggests that either algorithm could be considered as a first choice for collective document classification. Their advantage is increased by their ability to output classification confidences as probabilities, while minimum-cut and the local formulations only give absolute class assignments. We do not dismiss the iterative-classifier approach entirely. The most compelling point in its favor is its ability to unify content only and citation features in a single classifier. Conceptually speaking, such an approach should allow the two types of features to inter-relate in more nuanced ways. A case in point comes from our use of a fixed size context window to build a citation classifier. Future approaches may be able to do away with this arbitrary separation of features by training a local classifier to consider all words in terms of their impact on content-only classification and their relations to neighbors.

Probabilistic SVM normalization offers a convenient, principled way of incorporating the outputs of an SVM classifier into a collective classifier. An opportunity for future work is to consider normalization approaches for other classifiers. For example, confidence-weighted linear classifiers (Dredze et al., 2008) have been shown to give superior performance to SVMs on a range of tasks and may therefore be a better choice for collective document classification.

Of the three models trialled for local classifiers, context window features did best when measured in an oracle experiment, but citation count features did better when used in a collective classifier. We conclude that context window features are a more nuanced and powerful approach that is also more likely to suffer from data sparseness. Citation count features would have been the less effective in a scenario where the fact of the citation existing was less informative, for example, if a citation was 50% likely to indicate agreement rather than 80% likely. There is much scope for further research in this area.

References

Alina Andreevskaia and Sabine Bergler. 2008. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *ACL*, pages 290–298.

Mohit Bansal, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *COLING*, pages 15–18.

Mustafa Bilgic and Lise Getoor. 2008. Effective label acquisition for collective classification. In *KDD*, pages 43–51.

Mustafa Bilgic, Galileo Namata, and Lise Getoor. 2007. Combining collective classification and link prediction. In *ICDM Workshops*, pages 381–386. IEEE Computer Society.

Avrim Blum and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, pages 19–26.

Remco R. Bouckaert. 2003. Choosing between two learning algorithms based on calibrated tests. In *ICML*, pages 51–58.

Clint Burfoot and Timothy Baldwin. 2009. Automatic satire detection: Are you having a laugh? In *ACL-IJCNLP Short Papers*, pages 161–164.

Clint Burfoot. 2008. Using multiple sources of agreement information for sentiment classification of political transcripts. In *Australasian Language Technology Association Workshop 2008*, pages 11–18. ALTA.

Minh Duc Cao and Xiaoying Gao. 2005. Combining contents and citations for scientific document classification. In *18th Australian Joint Conference on Artificial Intelligence*, pages 143–152.

Fabien Cromires and Sadao Kurohashi. 2009. An alignment algorithm using belief propagation and a structure-based distortion model. In *EACL*, pages 166–174.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *ICML*, pages 264–271.

Zeno Gantner and Lars Schmidt-Thieme. 2009. Automatic content-based categorization of Wikipedia articles. In *2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 32–37.

Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, Lawrence Saul, and David Heckerman. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *ACL*, pages 257–260.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine*, 29:93–106.

David A. Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *EMNLP*, pages 145–156.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *ACL-IJCNLP*, pages 226–234.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *EMNLP*, pages 170–179.

Ben Taskar, Pieter Abbeel, and Daphne Koller. 2002. Discriminative probabilistic models for relational data. In *UAI*.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *EMNLP*, pages 327–335.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings ACM SIGIR*, pages 42–49.