The power of negative thinking: Exploiting label disagreement in the min-cut classification framework

Mohit Bansal
Dept. of Computer Science & Engineering
Indian Institute of Technology Kanpur
mbansal47@gmail.com

Claire Cardie and **Lillian Lee**
Dept. of Computer Science
Cornell University
{cardie,llee}@cs.cornell.edu

Abstract

Treating classification as seeking *minimum cuts* in the appropriate graph has proven effective in a number of applications. The power of this approach lies in its ability to incorporate label-agreement preferences among pairs of instances in a provably tractable way. Label *disagreement* preferences are another potentially rich source of information, but prior NLP work within the minimum-cut paradigm has not explicitly incorporated it. Here, we report on work in progress that examines several novel heuristics for incorporating such information. Our results, produced within the context of a politically-oriented sentiment-classification task, demonstrate that these heuristics allow for the addition of label-disagreement information in a way that improves classification accuracy while preserving the efficiency guarantees of the minimum-cut framework.

1 Introduction

Classification algorithms based on formulating the classification task as one of finding *minimum s-t cuts in edge-weighted graphs* — henceforth *minimum cuts* or *min cuts* — have been successfully employed in vision, computational biology, and natural language processing. Within NLP, applications include sentiment-analysis problems (Pang and Lee, 2004; Agarwal and Bhattacharyya, 2005; Thomas et al., 2006) and content selection for text generation (Barzilay and Lapata, 2005).

As a classification framework, the minimum-cut approach is quite attractive. First, it provides a principled, yet flexible mechanism for allowing problem-specific relational information — including several types of both hard and soft constraints — to influence a collection of classification decisions. Second, in many important cases, such as when all the edge weights are non-negative, finding a minimum cut can be done in a provably efficient manner.

To date, however, researchers have restricted the semantics of the constraints mentioned above to encode pair-wise "agreement" information only. There is a computational reason for this restriction: "agreement" and "disagreement" information are arguably most naturally expressed via positive and negative edge weights, respectively; but in general, the inclusion of even a relatively small number of negative edge weights makes finding a minimum cut NP-hard (McCormick et al., 2003).

To avoid this computational issue, we propose several heuristics that encode disagreement information with non-negative edge weights. We instantiate our approach on a sentiment-polarity classification task — determining whether individual conversational turns in U.S. Congressional floor debates support or oppose some given legislation. Our preliminary results demonstrate promising improvements over the prior work of Thomas et al. (2006), who considered only the use of agreement information in this domain.

2 Method

2.1 Min-cut classification framework

Binary classification problems are usually approached by considering each classification decision in isolation. More formally, let $X_{test} =$

Coling 2008: Companion volume – Posters and Demonstrations, pages 15–18
Manchester, August 2008

$\{x_1, x_2, \ldots, x_n\}$ be the test instances, drawn from some universe X, and let $C = \{c_1, c_2\}$ be the two possible classes. Then, the usual approach can often be framed as labeling each x_i according to some *individual*-preference function $\mathrm{Ind}: X \times C \rightarrow \Re$, such as the signed distance to the dividing hyperplane according to an SVM or the posterior class probability assigned by a Naive Bayes classifier.

But when it is difficult to accurately classify a particular x_i in isolation, there is a key insight that can help: knowing that x_i has the same label as an easily-categorized x_j makes labeling x_i easy. Thus, suppose we also have an association-preference function $\mathrm{Assoc}: X \times X \rightarrow \Re$ expressing a reward for placing two items in the same class; an example might be the output of an agreement classifier or a similarity function. Then, we can search for a classification function $c(x_i|X_{test})$ — note that all of X_{test} can affect an instance's label — that minimizes the total "pining" of the test items for the class they were not assigned to due to either their individual *or* associational preferences:

$$\sum_i \mathrm{Ind}(x_i, \overline{c}(x_i|X_{test})) + \alpha \times \sum_{i,j:c(x_i|X_{test})=\overline{c}(x_j|X_{test})} \mathrm{Assoc}(x_i, x_j),$$

where $\overline{c}(x_i|X_{test})$ is the class "opposite" to $c(x_i|X_{test})$, and the free parameter α regulates the emphasis on agreement information. Solutions to the above minimization problem correspond to *minimum s-t cuts* in a certain graph, and *if* both Ind and Assoc are non-negative functions, then, surprisingly, minimum cuts can be found in polynomial time; see Kleinberg and Tardos (2006, Section 7.10) for details. But, as mentioned above, allowing negative values makes finding a solution intractable in the general case.

2.2 Prior work discards some negative values

The starting point for our work is Thomas et al. (2006) (henceforth TPL). The reason for this choice is that TPL used minimum-cut-based classification wherein signed distances to dividing SVM hyperplanes were employed to define $\mathrm{Ind}(x, c)$ and $\mathrm{Assoc}(x, x')$. It was natural to use SVMs, since association was determined by classification rather than similarity — specifically, categorizing references by one congressperson to another as reflecting agreement or not — but as a result, negative association-preferences (e.g., negative distance to a hyperplane) had to be accounted for.

We formalize TPL's approach at a high level as follows. Let $\mathrm{Ind}': X \times C \rightarrow \Re$ and $\mathrm{Assoc}': X \times X \rightarrow \Re$ be *initial* individual- and association-preference functions, such as the signed distances mentioned above. TPL create two *non-negative conversion* functions $f: \Re \rightarrow [0, 1]$ and $g: \Re \rightarrow [0, 1]$, and then define

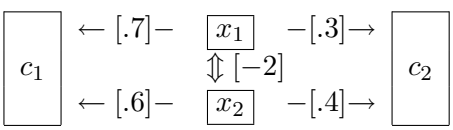
$$\begin{aligned} \mathrm{Ind}(x_i, c) &:= f(\mathrm{Ind}'(x_i, c)) \\ \mathrm{Assoc}(x_i, x_j) &:= g(\mathrm{Assoc}'(x_i, x_j)) \end{aligned}$$

so that an optimal classification can be found in polynomial time, as discussed above. We omit the exact definitions of f and g in order to focus on what is important here: roughly speaking, f and g normalize values and handle outliers[1], with the following crucial exception. While negative initial individual preferences for one class can be translated into positive individual preferences for the other, there is no such mechanism for negative values of Assoc'; so TPL resort to defining g to be 0 for negative arguments. They thus *discard* potentially key information regarding the strength of label *disagreement* preferences.

2.3 Encoding negative associations

Instead of discarding the potentially crucial label-disagreement information represented by negative Assoc' values, we propose heuristics that seek to incorporate this valuable information, but that keep Ind and Assoc non-negative (by piggy-backing off of TPL's pre-existing conversion-function strategy[2]) to preserve the min-cut-classification efficiency guarantees.

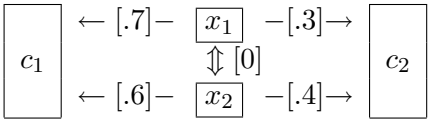
We illustrate our heuristics with a running example. Consider a simplified setting with only two instances x_1 and x_2; $f(z) = z$; $g(z) = 0$ if $z < 0$, 1 otherwise; and Ind' values (numbers labeling left or right arrows in the diagrams below, e.g., $\mathrm{Ind}'(x_1, c_1) = .7$) and Assoc' value (the -2 labeling the up-and-down arrow) as depicted here:



Then, the resulting TPL Ind and Assoc values are

[1] Thus, strictly speaking, f and g also depend on Ind', Assoc', and X_{test}, but we suppress this dependence for notational compactness.

[2] Our approach also applies to definitions of f and g different from TPL's.

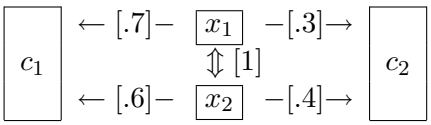


Note that since the initial -2 association value is ignored, $c(x_1|X_{test}) = c(x_2|X_{test}) = c_1$ appears to be a good classification according to TPL.

The *Scale all up* heuristic Rather than discard disagreement information, a simple strategy is to just scale up all initial association preferences by a sufficiently large positive constant N:

$$\begin{aligned} \mathrm{Ind}(x_i, c) &:= f(\mathrm{Ind}'(x_i, c)) \\ \mathrm{Assoc}(x_i, x_j) &:= g(\mathrm{Assoc}'(x_i, x_j) + N) \end{aligned}$$

For $N = 3$ in our example, we get

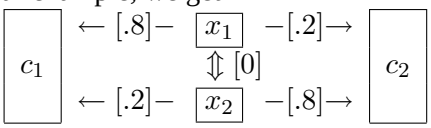


This heuristic ensures that the more negative the Assoc' value, the lower the cost of separating the relevant item pair (whereas TPL don't distinguish between negative Assoc' values). The heuristic below tries to be more proactive, *forcing* such pairs to receive different labels.

The *SetTo* heuristic We proceed through $x_1, x_2, \ldots$ in order. Each time we encounter an x_i where $\mathrm{Assoc}'(x_i, x_j) < 0$ for some $j > i$, we try to force x_i and x_j into different classes by altering the four relevant individual-preferences affecting this pair of instances, namely, $f(\mathrm{Ind}'(x_i, c_1))$, $f(\mathrm{Ind}'(x_i, c_2))$, $f(\mathrm{Ind}'(x_j, c_1))$, and $f(\mathrm{Ind}'(x_j, c_2))$. Assume without loss of generality that the largest of these values is the first one. If we respect that preference to put x_i in c_1, then according to the association-preference information, it follows that we should put x_j in c_2. We can instantiate this chain of reasoning by setting

$\mathrm{Ind}(x_i, c_1) := \max(\beta, f(\mathrm{Ind}'(x_i, c_1)))$
$\mathrm{Ind}(x_i, c_2) := \min(1-\beta, f(\mathrm{Ind}'(x_i, c_2)))$
$\mathrm{Ind}(x_j, c_1) := \min(1-\beta, f(\mathrm{Ind}'(x_j, c_1)))$
$\mathrm{Ind}(x_j, c_2) := \max(\beta, f(\mathrm{Ind}'(x_j, c_2)))$

for some constant $\beta \in (.5, 1]$, and making no change to TPL's definition of Assoc. For $\beta = .8$ in our example, we get

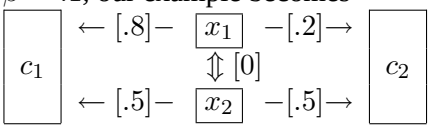


Note that as we proceed through $x_1, x_2, \ldots$ in order, some earlier changes may be undone.

The *IncBy* heuristic A more conservative version of the above heuristic is to increment and decrement the individual-preference values so that they are somewhat preserved, rather than completely replace them with fixed constants:

$\mathrm{Ind}(x_i, c_1) := \min(1, f(\mathrm{Ind}'(x_i, c_1)) + \beta)$
$\mathrm{Ind}(x_i, c_2) := \max(0, f(\mathrm{Ind}'(x_i, c_2)) - \beta)$
$\mathrm{Ind}(x_j, c_1) := \max(0, f(\mathrm{Ind}'(x_j, c_1)) - \beta)$
$\mathrm{Ind}(x_j, c_2) := \min(1, f(\mathrm{Ind}'(x_j, c_2)) + \beta)$

For $\beta = .1$, our example becomes



3 Evaluation

For evaluation, we adopt the sentiment-classification problem tackled by TPL: classifying *speech segments* (individual conversational turns) in a U.S. Congressional floor debate as to whether they support or oppose the legislation under discussion. TPL describe many reasons why this is an important problem. For our purposes, this task is also very convenient because all of TPL's computed raw and converted Ind' and Assoc' data are publicly available at www.cs.cornell.edu/home/llee/data/convote.html. Thus, we used their calculated values to implement our algorithms as well as to reproduce their original results.[3]

One issue of note is that TPL actually inferred association preferences between *speakers*, not speech segments. We do the same when applying *SetTo* or *IncBy* to a pair $\{x_i, x_j\}$ by considering the *average* of $f(\mathrm{Ind}'(x_k, c_1))$ over *all* x_k uttered by the speaker of x_i, instead of just $f(\mathrm{Ind}'(x_i, c_1))$. The other three relevant individual values are treated similarly. We also make appropriate modifications (according to *SetTo* and *IncBy*) to the individual preferences of all such x_k simultaneously, not just x_i, and similarly for x_j.

A related issue is that TPL assume that all speech segments by the same speaker should have the same label. To make experimental comparisons meaningful, we follow TPL in considering two different instantiations of this assumption. In *segment-based classification*, $\mathrm{Assoc}(x_i, x_j)$ is set to an arbitrarily high positive constant if the same speaker uttered both x_i and x_j. In *speaker-based classification*, $\mathrm{Ind}'(x_i, c)$ is produced by running

[3]For brevity, we omit TPL's "high-threshold" variants.

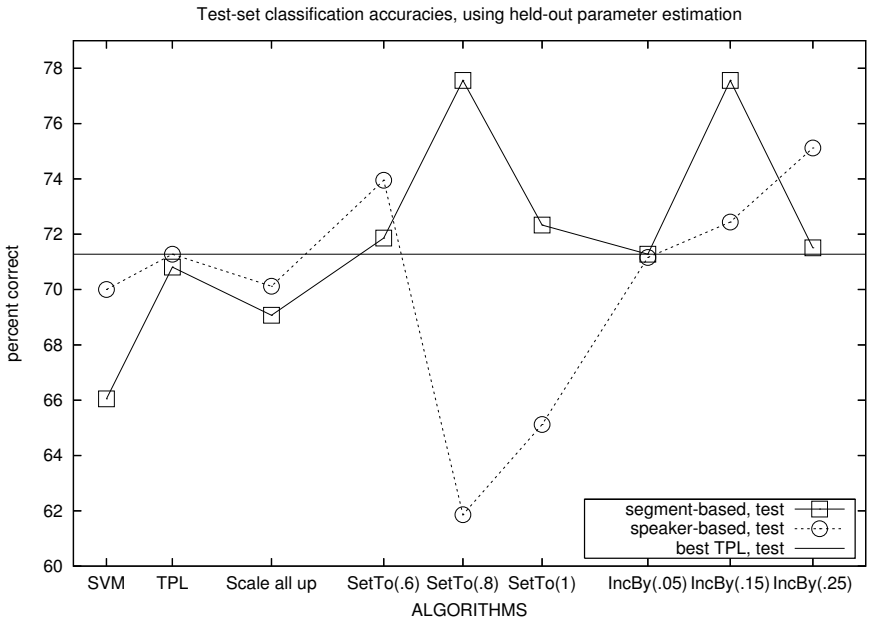


Figure 1: Experimental results. "SVM": classification using only individual-preference information. Values of β are indicated in parentheses next to the relevant algorithm names.

an SVM on the concatenation of all speeches uttered by x_i's speaker.

Space limits preclude inclusion of further details; please see TPL for more information.

3.1 Results and future plans

The association-emphasis parameter α was trained on held-out data, with ties broken in favor of the largest α in order to emphasize association information. We used Andrew Goldberg's HIPR code (http://avglab.com/andrew/soft.html) to compute minimum cuts. The resultant test-set classification accuracies are presented in Figure 1.

We see that *Scale all up* performs worse than TPL, but the more proactive heuristics (*SetTo*, *IncBy*) almost always outperform TPL on segment-based classification, sometimes substantially so, and outperform TPL on speaker-based classification for half of the variations. We therefore conclude that label disagreement information is indeed valuable; and that incorporating label disagreement information on top of the (positive) label agreement information that TPL leveraged can be achieved using simple heuristics; and that good performance enhancements result without any concomitant significant loss of efficiency.

These results are preliminary, and the divergence in behaviors between different heuristics in different settings requires investigation. Additional future work includes investigating more sophisticated (but often therefore less tractable) formalisms for joint classification; and looking at whether approximation algorithms for finding minimum cuts in graphs with negative edge capacities can be effective.

Acknowledgments We thank Jon Kleinberg and the reviewers for helpful comments. Portions of this work were done while the first author was visiting Cornell University. This paper is based upon work supported in part by the National Science Foundation under grant nos. IIS-0329064, BCS-0624277, and IIS-0535099, a Cornell University Provost's Award for Distinguished Scholarship, a Yahoo! Research Alliance gift, an Alfred P. Sloan Research Fellowship, and by DHS grant N0014-07-1-0152. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of any sponsoring institutions, the U.S. government, or any other entity.

References

A. Agarwal, P. Bhattacharyya. 2005. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. *ICON*.

R. Barzilay, M. Lapata. 2005. Collective content selection for concept-to-text generation. *HLT/EMNLP*, pp. 331–338.

J. Kleinberg, É. Tardos. 2006. *Algorithm Design*. Addison Wesley.

S. T. McCormick, M. R. Rao, G. Rinaldi. 2003. Easy and difficult objective functions for max cut. *Mathematical Programming*, Series B(94):459–466.

B. Pang, L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *ACL*, pp. 271–278.

M. Thomas, B. Pang, L. Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *EMNLP*, pp. 327–335.