Supplementary Material to
“Label Selection on Graphs”

Andrew Guillory
Department of Computer Science
University of Washington
guillory@cs.washington.edu

Jeff Bilmes
Department of Electrical Engineering
University of Washington
bilmes@ee.washington.edu

1 Ψ Function Submodularity and Supermodularity

Theorem 1. Ψ *is not submodular.*

Proof. Consider a binary weight graph consisting of a cycle of length 4 with vertices $1, 2, 3, 4$ along the cycle. For this graph

$$\Psi(\emptyset) = 0 \qquad \Psi(\{1\}) = \frac{2}{3}$$
$$\Psi(\{3\}) = \frac{2}{3} \qquad \Psi(\{1,3\}) = 2$$

So,

$$\Psi(\{1,3\}) - \Psi(\{3\}) = \frac{4}{3} > \Psi(\{1\}) - \Psi(\emptyset) = \frac{2}{3}$$

□

Theorem 2. Ψ *is not supermodular.*

Proof. Consider a binary weight graph consisting of a chain of length 4 with vertices $1, 2, 3, 4$ along the chain. For this graph

$$\Psi(\emptyset) = 0 \qquad \Psi(\{2\}) = \frac{1}{2}$$
$$\Psi(\{1\}) = \frac{1}{3} \qquad \Psi(\{1,2\}) = \frac{1}{2}$$

So,

$$\Psi(\{1,2\}) - \Psi(\{2\}) = 0 < \Psi(\{1\}) - \Psi(\emptyset) = \frac{1}{3}$$

□

2 Covering Objective Submodularity, Supermodularity, and Hardness

Krause et al. [2008] has an indirect proof that F' is not submodular. They give an example that shows the greedy algorithm can do arbitrarily bad when maximizing F' under a cardinality constraint. If F' were submodular the greedy algorithm would give a $(1 - 1/e)$ approximation for maximizing under a cardinality constraint. We give a direct proof that F' is not submodular for completeness. In our proof, we specifically show this holds even when W' is symmetric with ∞ along the diagonal (i.e. for the restricted case of W' given by our graph covering problem).

Theorem 3. *F' is not submodular.*

Proof. Consider the case with F' defined by

$$W' = \begin{bmatrix} \infty & 0 & 1 & 0 \\ 0 & \infty & 0 & 1 \\ 1 & 0 & \infty & 0 \\ 0 & 1 & 0 & \infty \end{bmatrix}$$

In this case

$$F'(\emptyset) = 0 \qquad F'(\{1\}) = 0$$
$$F'(\{2\}) = 0 \qquad F'(\{1, 2\}) = 1$$

So,

$$F'(\{1, 2\}) - F'(\{2\}) = 1 > F'(\{1\}) - F'(\emptyset) = 0$$

□

One might hope that instead F' is instead supermodular. The minimum of linear functions is concave, and modularity and supermodularity are often seen as discrete versions of linearity and concavity. This is not the case, however. We again give an example for when W' is symmetric and has ∞ along the diagonal.

Theorem 4. *F' is not supermodular.*

Proof. Consider F' given by

$$W' = \begin{bmatrix} \infty & 1 & 0 & 1 & 0 \\ 1 & \infty & 1 & 0 & 1 \\ 0 & 1 & \infty & 1 & 0 \\ 1 & 0 & 1 & \infty & 0 \\ 0 & 1 & 0 & 0 & \infty \end{bmatrix}$$

In this case,

$$F'(\{2\}) = 0 \qquad F'(\{1, 2\}) = 1$$
$$F'(\{2, 3\}) = 1 \qquad F'(\{1, 2, 3\}) = 1$$

So,

$$F'(\{1, 2, 3\}) - F'(\{1, 2\}) = 0 < F'(\{1, 2\}) - F'(\{2\}) = 1$$

□

We also show that the covering problem, specifically the special case of computing a dominating set in a binary weight graph, is NP-hard and hard to approximate by way of set cover. As mentioned, the connection to set cover is well known, but we do not have a reference for this proof.

Theorem 5. *Finding the smallest dominating set L in a binary weight graph is NP-complete. Furthermore, if there is some $\epsilon > 0$ such that a polynomial time algorithm approximates the smallest dominating set within $(1 - \epsilon) \ln(n/2)$ then $NP \subset TIME(n^{O(\log \log n)})$.*

Proof. To show the problem is NP-hard. We reduce the set cover problem to computing a minimal dominating set in a graph. Given k sets containing items from a domain of size m, we construct a graph of size $n = k + m$ with a vertex for each set and a vertex for each unique item. If item i is in set j, connect the vertex corresponding to i to the vertex corresponding to j in the graph. We now show that 1) any set cover can be mapped to a dominating set of the same size in the graph and 2) any dominating set in the graph can be mapped to a set cover without increasing the size of the set. From these two points, it follows that we can compute a minimal set cover by computing a minimal dominating set and then mapping this to a set cover of the same size. For 1), if a set of sets S is a set cover, then the set of vertices L corresponding to these sets is a dominating set. For 2), if a set of

	Spectral	k-cut	METIS	Ψ	α-Cover	Baseline
Digit1/251	3.36 (0.67)	49.52 (1.96)	2.74 (0.39)	1.84 (0.3)	2.24 (0.00)	**1.94 (0.36)**
Text/225	29.91 (1.92)	48.15 (5.21)	30.27 (1.25)	31.20 (7.67)	25.18 (0.00)	**21.53 (1.34)**
BCI/60	47.07 (3.15)	50.37 (0.31)	**46.86 (2.08)**	49.37 (2.20)	49.12 (0.00)	48.44 (2.65)
USPS/225	5.92 (1.14)	23.38 (23.83)	6.68 (0.84)	4.20 (2.46)	4.24 (0.00)	**3.95 (1.01)**
g241c/50	45.15 (2.80)	50.03 (0.1)	41.14 (2.77)	51.31 (0.24)	**37.10 (0.00)**	46.53 (4.47)
g241d/56	40.72 (2.48)	50.03 (0.03)	35.43 (2.6)	50.21 (0.14)	**34.70 (0.00)**	45.29 (3.79)

Table 1: Error rate mean (standard deviation). Label counts were set to the size of the computed α-cover with α set to the minimum weight in the graph.

vertices L is a dominating set we construct a set cover S of size at most $|L|$ by mapping items in L to sets in S. For every vertex in L corresponding to a set, include that set in S. For every vertex, in L corresponding to an item, include an arbitrarily chosen set containing that item in S.

To show the hardness of approximation result, note that this reduction is approximation preserving with respect to m up to a factor of 2 assuming that $k < m$. This is the case because the optimal neighborhood cover size is the same as the optimal set cover size and if $k < m$ then $n < 2m$. In this case, a $(1-\epsilon)\ln(n/2)$ approximation for the neighborhood cover problem would give a $(1-\epsilon)\ln m$ approximation for the original set cover problem. Feige [1998] shows there is no polynomial time $(1-\epsilon)\ln m$ approximation algorithm for set cover unless $NP \subset TIME(n^{O(\log\log n)})$ and in fact shows this using only set cover instances for which $k < m$. $\square$

3 Additional Experiments

We also experimented with a method motivated by the graph covering bound. In particular, we tried computing an α-cover and then predicting using majority vote. A drawback of this method is that it requires at least as many labels as the minimum dominating set (any α-cover is at least this large). The method also does not take in a label count but rather an α value although it is possible to find an α value by binary search. In our experiments we set α to be the minimum non-zero weight in the graph and use the size of this α-cover as a label budget for the other methods. We found that even using this method, which approximates the smallest α-cover, the number of labels required were such that none of the label selection methods consistently beat the baseline method. Table 1 shows these results. We also note that the α-cover method is likely to be outlier sensitive like the Ψ function bound on which it is based. In particular, we are forced to cover every point, including very weakly connected points.

We also tried a few modifications to our algorithms. In particular we tried using the method of Bengio et al. [2006] for prediction in conjunction with our label selection methods (i.e. instead of labeling each cluster with the majority label or using min-cuts). For the clustering methods, we also select a cluster center instead of a random point in each cluster by selecting the vertex i in a cluster S that maximizes $\sum_{j \in S} W_{i,j}$. For the Ψ function and baseline methods, we also exclude from label selection vertices i such that $\sum_j W_{i,j} \leq .1/n \sum_{i,j} W_{i,j}$ in a simple attempt to avoid outliers, but we found this rule only removed a significant number of points on the USPS data set. We found these modifications improve performance in the 100 label case, although they do not give uniformly better performance in general. Interestingly, this simple outlier rejection method sometimes removed very few or no vertices and other times hurt the performance of the baseline suggesting outlier rejection is difficult in general.

Our error rates for the baseline method are different than published results [Chapelle et al., 2006]. Some of this difference is explained by changes in experiment setup: Chapelle et al. [2006] use a fixed set of 12 train/test splits for both the 10 and 100 label cases while we use 1000 and 100 random splits for the 10 and 100 label cases respectively. Especially for the 10 label case, the variance in error rates over multiple splits can be large, and as such, the difference between error on a random selection of 12 splits and a random selection of 1000 splits can often be large. Also, Chapelle et al. [2006] force every split to contain a point from both classes. We do not force our random splits to have this property, although we expect most random splits will. Table 3 shows that using our graph parameters with the splits used by Chapelle et al. [2006] we get similar results with a few exceptions. We attribute these exceptions (the synthetic g241c / g241d data sets and the Digit1 data set for 10

	Spectral*	METIS*	Ψ*	Baseline*	Baseline
Digit1/10	8.88 (3.15)	**3.96 (0.00)**	11.20 (9.50)	21.10 (15.83)	20.90 (15.67)
Text/10	50.04 (1.10)	50.00 (0.00)	49.97 (0.17)	**45.76 (8.05)**	45.91 (7.96)
BCI/10	**49.09 (2.20)**	51.03 (0.00)	49.75 (1.09)	50.17 (1.36)	50.12 (1.32)
USPS/10	12.22 (3.23)	**9.60 (0.00)**	10.84 (1.49)	18.32 (3.09)	15.87 (4.82)
g241c/10	**45.34 (6.45)**	50.07 (0.00)	50.29 (0.07)	47.32 (5.13)	47.26 (5.19)
g241d/10	49.44 (1.97)	**44.90 (0.00)**	49.96 (0.08)	48.43 (3.53)	48.46 (3.39)
Digit1/100	2.27 (0.25)	2.57 (0.00)	**2.28 (0.52)**	2.60 (0.86)	2.57 (0.67)
Text/100	26.11 (2.42)	30.57 (0.00)	**25.66 (3.39)**	26.37 (3.40)	26.82 (3.88)
BCI/100	**47.04 (2.17)**	49.00 (0.00)	47.04 (6.42)	47.51 (2.82)	47.48 (2.99)
USPS/100	4.10 (0.88)	4.00 (0.00)	**2.66 (0.24)**	13.30 (3.65)	6.33 (2.46)
g241c/100	36.16 (3.33)	**31.86** (0.00)	52.56 (0.41)	43.48 (4.01)	42.86 (4.50)
g241d/100	34.33 (3.55)	**26.71** (0.00)	38.51 (0.68)	41.96 (4.67)	41.56 (4.34)

Table 2: Error rate mean (standard deviation). Algorithms marked with a * use heuristic modifications discussed in the text.

	Our Implementation	Published
Digit1/10	18.83	9.80
Text/10	42.33	40.79
BCI/10	50.49	50.36
USPS/10	13.79	13.61
g241c/10	47.82	39.96
g241d/10	48.55	46.55
Digit1/100	2.26	3.17
Text/100	26.33	30.54
BCI/100	46.92	46.22
USPS/100	5.74	6.36
g241c/100	43.28	22.05
g241d/100	40.80	28.20

Table 3: Average error rate using train/test splits from [Chapelle et al., 2006]

labels) to graph parameter choices and differences in implementation. We mention differences in implementation because even using a completely connected graph with $k_1 = 10$, which are the settings reported by Chapelle et al. [2006] for the 10 label case, we do not get the same results, and in fact using these parameters increases the error in several cases as compared to the parameters we found. One change we found significantly improved results on most data sets is specifying by hand a $.5, .5$ prior for the class mean normalization step instead of using a prior estimated from data. However, this is arguably introducing additional domain knowledge.

Finally, we tried comparing against two additional baseline methods. We use greedy submodular function maximization to select the labeled set using two different objectives. We use a facility location objective function

$$F(S) = \sum_{i \in V} \max_{j \in S} W_{i,j}$$

and a graph cut objective

$$F(S) = \sum_{i \in (V \setminus S)} \sum_{j \in S} W_{i,j}$$

Table 4 shows that these methods perform reasonably, but our METIS based clustering method is usually better in the 10 label case.

References

Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2006.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT press, 2006.

U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 1998.

	METIS	Facility Location	Graph Cut
Digit1/10	4.93	**3.42**	**3.42**
Text/10	**34.76**	50.20	50.34
BCI/10	**49.68**	51.03	50.51
USPS/10	**8.15**	20.13	20.13
g241c/10	**29.18**	50.07	38.05
g241d/10	**22.57**	49.93	49.93
Digit1/100	3.24	**2.21**	3.86
Text/100	**32.57**	43.79	43.86
BCI/100	45.35	47.33	**44.00**
USPS/100	**9.28**	21.43	21.43
g241c/100	37.47	33.07	**30.43**
g241d/100	35.96	33.5	**28.00**

Table 4: Error rate means comparing our METIS clustering method to submodular function maximization

A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *JMLR*, 2008.