Mike Jin  
CPSC 490  
Project Proposal  
Advisor: Daniel Spielman

Improving on Graphs Fitted to Vector Data

**Introduction**

Most real-world quantitative and categorical data can be represented as vectors. There exist many methods for obtaining useful information from such data, e.g., different types of clustering to find groups of data points more related to each other than to the others, and regression to obtain parameters that describe the data set as a whole. Recently, researchers have found fitting graphs to data to be both efficient and informative for solving some classification, regression, and clustering problems. In these graphs, each vector represents a node, and it is the determination of edges and edge weights that is of interest. One important practical motivation in solving these kinds of problems is semi-supervised machine learning: given a set of data and labels assigned to a subset of the data by a skilled human, one wishes to obtain accurate labels for the unlabeled data.

Initial efforts at fitting graphs to vector data introduced parameters that could be tuned to obtain the desired fit of the graph: for example, defining edges between nodes that were less than $k$ distance apart, or defining edges between each node and its $m$ nearest neighbors. Subsequently, attempts were made to combine these different graphs in an optimally weighted average. Recently, however, theoretical motivations produced a method which did not require any externally imposed parameters. This method by Daitch, Kelner, and Spielman was shown to generally outperform existing semi-supervised learning algorithms for regression and classification when applied to real-world data sets.

Daitch et al.'s method constructs an undirected weighted graph on a given set of vectors $x_1, \ldots, x_n$, where the weight on edge $(i, j)$ is given by $w_{i,j} \geq 0$ (0 denoting nonexistence of the edge) and the weighted degree of node $i$ is given by $d_i = \sum_j w_{i,j}$. The fitted graph is determined by minimizing the sum of squares of distances, weighted by degree, from each node to the weighted average of its neighbors:

$$\min_w \sum_i \left\| d_i x_i - \sum_j w_{i,j} x_j \right\|^2$$

This minimization problem may be solved using convex quadratic programming. It has been shown that, using this method, the graphs produced are sparse and that graphs for vectors in $\mathbb{R}^2$ are always planar. Sparsity of the graphs is of practical interest because dense graphs are both less computationally tractable and less interesting because they begin to resemble complete graphs. Furthermore, it has been suggested that the average degree of a fitted graph is a useful measure of the dimensionality of the vectors.
It is possible that this method may be improved by modifying the function to be minimized. Two modifications will be investigated, both separately and in combination. The first is to remove the square on the norm. Data sets regularly contain outliers which add noise and potentially negatively influence the ability of machines to perform classification and/or regression accurately. Squaring the norm of each term in the sum would magnify these effects by increasing the relevance of outliers. The second is to change the Euclidean norm to a 1-norm, or Manhattan norm. This would again reduce the influence of coordinate-wise outliers. In addition, this modification would change the minimization problem from a quadratic program to a linear program, which is solved more quickly by computers.

**Scope of the project**

The project will begin with an experimental investigation of the effects of using the modified objective function. Properties of the generated graphs such as sparsity and planarity will be investigated empirically. We conjecture that graph sparsity will still hold even with the modifications of the objective function but that planarity will likely no longer hold. Also possible for study is whether planarity still holds for some data sets, and whether these data sets or their generated graphs have interesting properties. Additionally, we will investigate the success of graphs generated using the modified objective functions in performing semi-supervised classification and regression tasks on real-world data sets as compared to the original objective function and existing semi-supervised learning algorithms. Success will be determined by looking at both running time and standard measures of learning performance, e.g. percent classification error and regression mean-square error for the two types of experiments, respectively.

We will also experimentally investigate the possibility of determining irrelevant coordinates in data sets. This would be useful in reducing the dimensionality of the problem and simplifying regressions. The general approach would be weight the coordinates of data vectors by dimension and noting the effects on classification and regression success due to perturbations of the weights: essentially like a derivative of machine learning success with respect to coordinate weight. Coordinates determined to have little effect on learning success can be discarded.

Proofs or counterexamples will be attempted based on experimental results. If our conjectures are correct, then a proof for the sparsity of generated graphs using one or more of the modified objective functions will be attempted; otherwise, we will attempt to find a counterexample with a dense generated graph. Based on checking for existence of graph planarity and provided that certain properties hold for some subset of the planar graphs, proofs or counterexamples will be attempted accordingly.

Whether the modifications to the objective function improve runtime or machine learning performance on data sets overall or not, it is of practical interest to study what kinds of data sets it does perform well on, whether relative to itself, or to the original objective function or other existing algorithms. Toward this aim, it may prove useful to generate data sets with desired properties and, by comparing the performance on these data among the machine learning methods, build confidence in an
empirical description of the value of this method on different kinds of data, which would be useful for someone evaluating the relative merits of different algorithms for a particular data set.

Finally, it is of obvious practical interest to optimize the runtime for one of these methods using a modified objective function or for the original quadratic program. All code for the original program was written in MATLAB and used a package for convex programming called CVX. Additional optimizations, either based on basic properties of the input or for any input, may be determined and implemented to decrease runtime. Bypassing CVX, a more specialized convex programming solver may be written to outperform it in runtime as well.

**Deliverables**

- Experimental results and analysis
  - Comparison of runtime performance to previous algorithms
  - Comparison of classification and regression tasks to previous algorithms
  - Occurrence of graph sparsity and/or planarity
  - Ability to determine irrelevant coordinates
- Code for new linear programs and/or modified quadratic program
  - Optimized programs, with or without bypassing CVX
- Theorems based on findings in experimental results
  - Proof/counterexample for sparsity/planarity of generated graphs
- Final written report