

15781 Project Ideas

1. Try to track down low level data (e.g. by county) from the recent US election combined with census information about the make-up of each county and then use one of more of the machine learning tools from class to create a model that's best able to predict the outcome for that county. This project is one where the main hard part concerns tracking down the data and creative ways to manipulate it. *AM*
2. Investigate the extent to which multiple-step lookaheads can improve decision-tree performance. Try your method out both on real data and artificial data designed to fool greedy decision trees. *AM*
3. Simple language modeling: try to build a predictive model that can predict the next letter in text based on previous letters or other features about the recent sentence. Find out whether this can in principle get better compression than ZIP. Or do the same at the level of full words. Or for the Genome. *AM*
4. Create an educational animation of SVM learning for 2-d data, perhaps entered by the use by a mouse, for some or all of the cases of {Linear or Radial Kernel-based} SVM on {Separable data or noisy data} (Obviously this would be of great benefit to future generations of students!) *AM*
5. Comparison of algorithms: Throughout the course, we've been discussing various algorithms and their properties, but only on occasion have we dealt with these algorithms with real sets of data. Often times, algorithms don't work like expected and algorithms may need to be adapted or modified to better fit the assumptions inherent in the problem or the available data. What work needs to be done to adapt a model to an interesting set of data that you've found? How do various algorithms perform on the same set of data? What are the properties of the various algorithms that exhibit such performance? *YQ*
6. Clustering input attributes, designing clustering metrics. *YQ*
7. The choice of the kernel function in SVMs: The kernel function in SVMs defines how examples are to be compared. How do we choose the kernel function? How could we adjust the kernel function if we thought it should have a particular form? Can you adapt/design a kernel function to a specific problem you are interested in solving? *YQ*
8. Detecting abnormal/novel examples in a stream of data. *YQ*
9. Learning Routines - Imagine we're given some data concerning a person's location at various times throughout the day (for a month, say). How could we use some of the ideas looked at in class (such as EM, GMM's, etc) to learn a few daily routines of the person, then use these routines to predict where the person is going on a new day, given some partial information (like the location of the person in the morning). *DF*
10. Predicting NFL/NBA outcomes - Grab a bunch of statistics that you feel are useful for predicting the outcome of an NFL game (perhaps from last season). Investigate which of these can be treated independently and which should be jointly considered to get the best prediction accuracy over the games so far this season. Can you tell us if the Steelers will win the Superbowl? *DF*
11. Intersection Detection/Recognition - Part of the CMU Mine Mapping Project entails locating intersections inside mines. We have software that can do this quite robustly. However, it would be very useful to be able to recognize intersections that have been

- previously visited. Given a 3D point cloud for each intersection already visited and a new 3D point cloud for a new intersection, investigate how we could find which (if any) of the previously visited intersections this new one corresponds to, along with some confidence measure. Extra for experts would be to reduce the dimensionality of the stored intersections so that we didn't have to store or compare full 3D point clouds. *DF*
12. Do unsupervised clustering of ascii files on a computer, and allow a user to do a search for 'similar based on contents' files on the computer. *ML*
 13. Investigate ways of fitting a Bayes net into training data. In particular, one can experiment with the simple and efficient greedy algorithms for the construction of decision trees and see how they apply here. *ML*
 14. We have one or two large files with a 3D map of environment constructed from robot laser scans. Few things to try to do with it: (a) Learn a classifier of surfaces (e.g., asphalt, gravel, tree-like surfaces, building faces, ...). (a) Fit surface segments into the data to represent the data more compactly. *ML*
 15. Try to detect an unusual activity on your computer and thus possibly detect a virus. It is a question as to what would be the input data. Supposedly, one could use a system log file, but this needs further investigation. *ML*

AM – Andrew Moore

DF – Dave Ferguson

ML – Max Likhachev

YQ – Yanjun Qi