**CSC529: Advanced Data Mining**

**Final Project: instructions, data sources, and examples of methodologies for the projects**

1. How to choose and work on the final project

It is helpful to initially focus on a **specific problem domain** that you find important and exciting. Consider what **the fundamental task** is that needs to be solved and think about how it might map onto, e.g., classification or clustering. Catalogue the **types of data** that are available and consider how these might be exploited. What are **the features** that will help your algorithm make decisions or predictions? Don’t be afraid to make assumptions that help establish an abstraction. Prefer abstractions and assumptions that may generalize beyond the immediate task at hand.

The next step is critical: define **quantitative metrics for success on held-out test data**. Classification accuracy? Area under the ROC curve? Predictive log probability? If you are focused on resource constraints, then perhaps your metrics will be curves that measure prediction as a function of, e.g., memory usage or CPU time. If possible, consider several possible metrics.

Once you have laid out the task, data and metrics, try to apply the dumbest, **simplest possible algorithm first.** Do not immediately try your new fancy idea. If you have a classification problem,try a single classifier first. Try things from the literature that seem like they would be applicable.Establish baselines for comparison that are honest attempts at doing well on the problem. After you’ve done this, you’ll have a much better idea of what your algorithm is capable of. You may also learn that certain approaches are difficult to beat. What does that mean? Maybe you need to focus on extracting features rather than a fancy classifier. Applying simple things first will help you understand where the frontier of the problem lies and help you determine whether your abstraction actually provides the information that you require for the task. In the end, it may be difficult to make a methodological contribution. If you have taken a problem-driven approach, however, then you have still done useful research by **improving our understanding of how data mining algorithms behave when applied to new problems**.

1. Some Sources for Datasets

* KDNuggets: <http://www.kdnuggets.com/datasets/>
* Datasets from competitions:
  + KDD cup: <http://www.kdnuggets.com/datasets/kddcup.html>
  + Kaggle (http://www.kaggle.com/competitions) – Sign up for an account to download the data.
* Yahoo datasets: <http://webscope.sandbox.yahoo.com/index.php>

1. Project Proposal

Proposal should be 1 page and single spacing. It should contain

* + Project title, student/team name, section (in-class vs online)
  + Description/Introduction Section: give motivation; describe the problem
  + Proposed Work Section: explain the idea; explain proposed approaches and methodology, propose possible alternative approaches if original plan does not work, explain experiments to be performed, data to be used.
  + References: provide at least 3 related references (excluding the textbooks and papers discussed in class)

1. Examples of Main Methodologies for Final Projects
2. Comparisons of probabilistic classifiers
   1. Across different types of classifiers
   2. Across different datasets
   3. Across different performance metrics
3. Comparisons of sampling techniques
   1. Over-sampling approaches
   2. Under-sampling approaches
4. Cost-sensitive learning
5. Comparison of ensemble of classifiers
   1. Across different types of classifiers
   2. Across different datasets
   3. Across different output combination schemes
6. Identify the relationships between initial performance, number of cases, and number of classifiers within an ensemble
7. Analyze the effects of prior outlier removal on boosting
8. Use AdaBoost to identify instances that are “either mislabeled or that are inherently ambiguous and hard to categorize” for a certain application.
9. Selective classifiers

The idea is to allow classifiers to abstain on hard instances, passing them to down-stream classifiers that may have more context available.

Example of an application: <http://research.microsoft.com/pubs/217153/SelectiveClassifiersTechReport.pdf>

1. Study the effect of using weights in the learning algorithm:
   1. How can k-NN algorithm handle weights implicitly
   2. How can DTs algorithm can handle weights implicitly
2. Multi-class multi-instance classifiers
3. Compare linear versus non-linear classifiers
4. Compare linear versus non-linear feature representations
5. Apply/compare hidden Markov models
6. Apply Bayesian networks to learn relationships between variables and make probabilistic inferences.
7. Ensemble of clustering approaches

Any of these ideas can be tested on your own data or any publically available datasets.