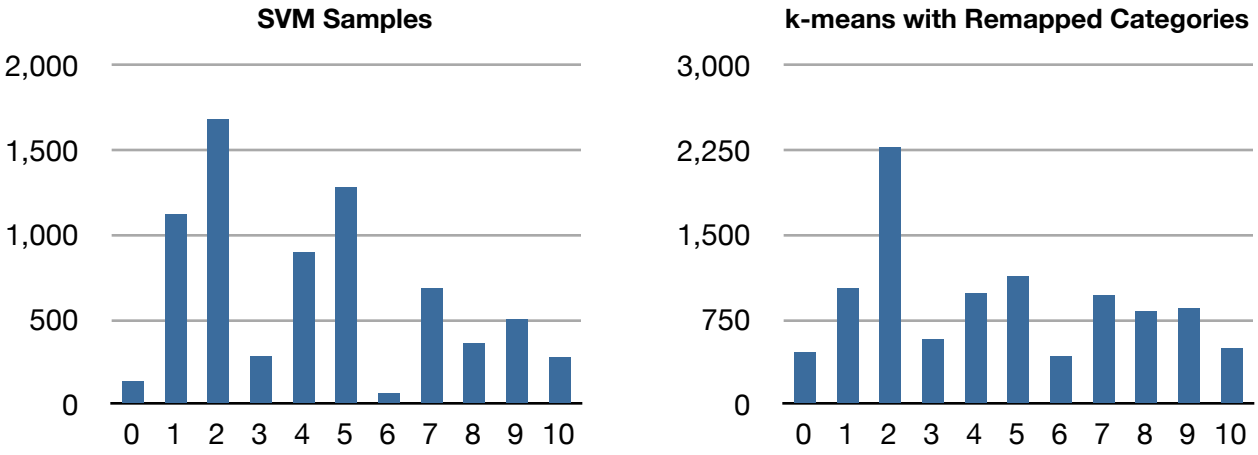
Report: Search Engine Final, Part 1

Alex Berke and Matt Patenaude

Explain how one can compare the result of the SVM algorithm with the *k*-means algorithm.

At first blush, comparing the results of SVM and *k*-means is difficult: *k*-means does not preserve category label information, it only identifies naturally occurring clusters among data points. However, if *k*-means and SVM share any kind of accuracy, they should both produce categorization histograms with peaks that follow a similar shape. Thus, to compare *k*-means and SVM, one solution is to sort the both sets of output by the number of documents in each category in ascending order, then adopt the category labels from SVM for *k*-means. When re-sorted into ascending label order, the histograms (for the testing data only) look something like this:



As you can see, though the peaks are somewhat more pronounced with SVM, the histograms follow the same general shape, which indicates that the bars likely pick out the same categories. Using this information, the efficacy of the two algorithms can be more easily compared.

Briefly explain the pros and cons of using SVM and *k*-means algorithms.

SVM has a couple of important advantages over *k*-means: for one, it's a good deal faster; for another, it allows you to preserve category labels, which makes it potentially better suited to applications where you want to classify documents against a user-facing scale (rather than just, e.g., to enhance search results). However, SVM also requires a corpus of training data which can be difficult to procure, and the training data's signal-to-noise ratio seems to have a profound effect on the results.

Meanwhile, *k*-means does not depend on such carefully composed data, and judging by some of our results, appears to be more accurate. It's also useful for identifying categorization that may be latent, which is excellent for situations where you're trying to identify patterns without modeling them yourself beforehand in training data. It is, however, slower, and as mentioned before, assigns its own unique labels to the clusters it identifies.