Autocategorization

Andrew Carnes[1]

[1] *University of Florida*

Abstract

In order to maximize the chance of discovering a certain effect, many scientific analyses extract subsets of data from the main data set that are sensitive to the effect. If the effect exists, the data in these sensitive subsets is expected to differ significantly from the null hypothesis. There are many ways to extract subsets, but an optimal method should minimize the global p-value on the null hypothesis given the effect. Call the orthogonal subsets of data categories and the selection of subsets categorization. An optimal categorization enhances the power of the experiment and reduces the data needed for discovery. By specifying a metric corresponding to the expected p-value (given the effect), the optimum categorization may be automated using a decision tree. This paper describes the autocategorization algorithm.

1. Introduction

Many scientific analyses attempt to rule out a null hypothesis and declare a discovery. In this context, seeing a large enough discrepancy in the data from the null's prediction leads to a low p-value, the rejection of the null, and a discovery. With a model for the null and another for the expected data, the expected p-value can be quantified. An analysis with a lower p-value represents more convincing evidence to rule out the null, and therefore an analysis with a lower expected p-value is considered more sensitive. In addition, a more sensitive analysis requires less data to rule out the null hypothesis. For both of these reasons, optimizing the sensitivity is critical for any scientist.

The expected hypothesis and the null hypothesis determine probability density functions (PDFs) in a many dimensional feature space. However, data sparesly populates high dimensional spaces, and fits in high dimensions are difficult or untrustworthy in practice, so the PDFs are usually compared over a single discriminating feature - sometimes a few - to determine the p-value. Unfortunately, upon reducing to a lower dimensional PDF comparison, the sensitivity of the analysis is often reduced: feature space regions with large discriminating power are combined with those of high probability but low sensitivity.

To regain sensitivity, the regions of feature space with large discriminating power need to be extracted. In this way, it makes sense to divide the feature space up into separate regions, called categories, and compare the low dimensional PDFs within the categories to maximize the sensitivity. By analytically specifying a metric corresponding to the expected p-value the optimum categorization can be automated. This is separate problem from regression or classification where fit values are applied to different regions of space as to minimize some loss function. The goal is simply to divide up feature space to maximize the statistical sensitivity.

2. The Autocategorizer

This section explains the autocategorization algorithm (autocategorizer) in terms of a binned counting experiment along a single feature x_{pdf}. The remaining features are labeled x_1, x_2, etc. Let $B_{c,i}$ represent the amount of data predicted in the i^{th} bin of category c according to the null hypothesis H_o. Let $B_{c,i} + S_{c,i}$ represent the amount of data predicted by the expected hypothesis, H_1. B is the background denoting no discovery. S is the signal denoting the difference between H_1 and H_o.

$$\text{Net Significance} = \sum_{c,i} (y_{c,i} - \mu_{c,i})^2 / \sigma^2_{c,i} = \sum_{c,i} S^2_{c,i} / B_{c,i} \tag{1}$$

where i labels the bin, c denotes the category, y is the amount of data expected according to H_1, $\mu_{c,i}$ is the amount in a bin expected according to H_o, and

$\sigma_{c,i}$ is the standard deviation in a bin given H_o. When the net significance is large, the difference between H_1 and H_o is large compared to the expected fluctuations from H_1. Note that a larger net significance implies a lower p-value, and that maximizing the net significance will minimize the p-value. In this case the sensitivity metric is a χ^2 variable, where $y_{c,i}$, $\mu_{c,i}$, and $\sigma_{c,i}$ have been approximated by the Poissonian values in each bin. Note that other sensitivity metrics may be used as long as they track the p-value appropriately.

With the net significance acting as the sensitivity metric, a decision tree[1] can split up the feature space as to maximize the metric. The decision tree algorithm greedily builds the optimum categorization by recursively splitting feature space regions into two using hyperplanes. On the first iteration, the

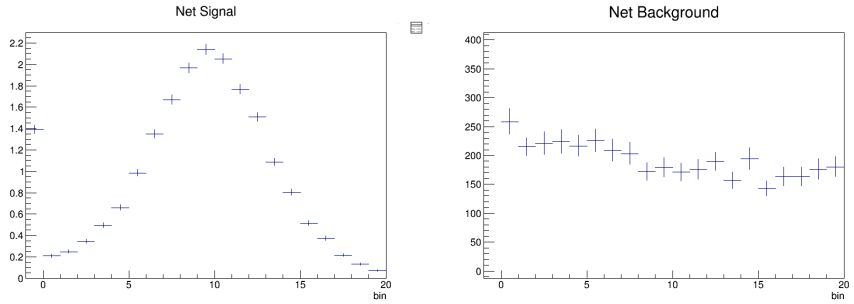

Figure 1: An example of the binned signal and background necessary for this significance metric. The binning keeps the most sensitive bins from being drowned out by the others.

autocategorizer calculates the net significance for the inclusive set of all events. The algorithm then searches over the inclusive set, checking all possible split values of the first feature, x_1. Events with x_1 values less than the split go in one candidate category, and those with x_1 values greater than or equal to the split value go into the other candidate category. For every split candidate, the algorithm calculates the net significance in the two categories delineated by the split value. The x_1 split value that provides the largest gain in significance over the inclusive set is stored. The gain is defined in the equation below where c1

and c2 are the prospective categories created from c by splitting on the feature.

$$\text{Gain} = \sum_i S^2_{c1,i}/B_{c1,i} + \sum_i S^2_{c2,i}/B_{c1,i} - \sum_i S^2_{c,i}/B_{c,i} \tag{2}$$

The autocategorizer then searches over the second feature, x_2, and stores the split value that provides the largest gain in significance. The process is repeated for all of the remaining features.

The algorithm chooses to split at the feature value with the largest gain, creating two categories from the inclusive set of events. At the next iteration, the autocategorizer repeats the procedure for the two new categories and chooses to split the category that provides the most gain. This process continues, each time greedily choosing to split the category with the most gain, until the number of categories desired is reached.

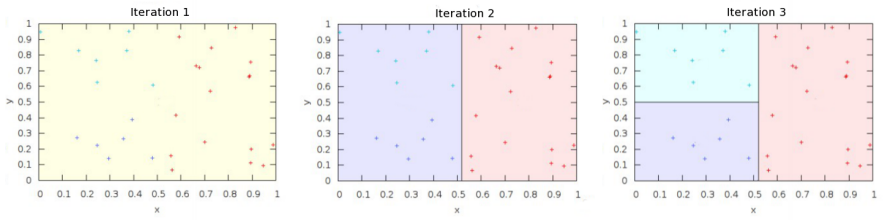

Figure 2: An example of the categorization process for features x,y and three categories. The autocategorizer chooses x=0.52 for the first split and y=0.50 for the second. The colored crosses represent the events that should be grouped together for optimum sensitivity. After three iterations, the categorizer correctly groups those events.

3. Conclusions

The autocategorizer algorithm automatically maximizes the chance to discover an effect of interest in a statistical analysis by extracting sensitive regions from a multidimensional feature space. The algorithm has been used at the Compact Muon Solenoid (CMS) experiment at CERN to set limits on the Higgs particle's rate of decay to two muons [2]. In the Higgs to dimuons analysis, the autocategorizer improves the upper limit on the rate of decay by 15% compared

to human expert categorization with the same simulated data and the same features. The improvement is equivalent to collecting 32% more data.

References

1. Breiman L, Friedman J, Olshen R, Stone C. Classification and Regression Trees. Monterey, CA: Wadsworth and Brooks; 1984.

2. CMS Collaboration . Search for the standard model Higgs boson decaying to two muons in pp collisions at $\sqrt{s} = 13$ TeV 2017;URL: `http://cds.cern.ch/record/2292159`.