Project 3: Predicting Mechanisms of Action in drug development arrays

Pablo Freyria Dueñas

May 10, 2021

Abstract

Previously, drug development was inspired by natural products or traditional remedies, which required little understanding on the biological mechanisms driving the pharmaceutical properties of drugs. Nowadays, drug development is made in a targeted way, driven by understanding the biological processes of a disease; this activity is often referred to as Mechanism of Action (MoA). This report aims at developing an algorithm that predicts the probability of 206 MoAs based on experiment settings (drug, dose, duration), 772 gene expressions and responses in 100 different human cells types.

1 Data and problem setting

Full data consists of $23,814$ experiments for which the following variables were measured: id, indicator of compound or control, duration of exposure,dosage (high/low), expression level of 772 genes and cell viability (% of cells alive after experiment) of 100 cell types. As target values, 206 MoA per experiment were labeled as 1-activated or 0-unactive. It is worth noticing that data had no missing values and that minimum processing was necessary: id was dropped as rows ID in features table matched those in targets, experiments with control vehicle ($n = 1,866$) were dropped off as no MoA was activated and dose was converted to numeric value (1 high; -1 low).
A key characteristic of the problem is that MoAs are not mutually exclusive and there can be multiple MoAs active per experiment. As descriptive metrics of MoA activation: 7.5K(34%)[1] had no activation, 12.5K (57%) of experiments activated one MoA, 1.5K (7%) had two MoAs active, and only 6 (0%) had 7 MoAs active. On the other hand, each MoAs can be considered as a "rare" event, in terms that the most common was active on 832 (3%) of experiments, while the least common was active on 1 (0%).(see figure 1)

[1]Throughout the report only experiments with active compound are considered

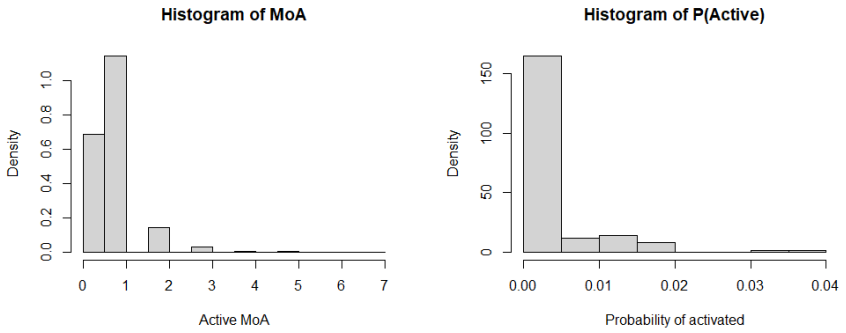

Figure 1: Histograms of MoA activation

As exposed above, for each experiment there are 206 binary classification problems with 874 predictors each. Problem was approached with multiple algorithms described in the following section and their performance was assessed with the log loss score function:

$$L(Y, P) = \frac{1}{NM} \sum_{i,j} -y_{i,j} log(p_{i,j}) - (1 - y_{i,j}) log(1 - p_{i,j}) \tag{1}$$

Where, Y is the target value (0 or 1), P the predicted probabilities, N the number of experiments (indexed by i) and M the total number of MoAs to classify. As a benchmark, and noting that probability of being active is low, using $P = 0$ yields an expected loss (risk) of 0.026, while using the "naive" probability of each MoA being active yields a risk of 0.022. It is worth highlighting that the rareness of the outcome makes this problem particularly difficult: a constant guess for all experiments has low risk.

2 Algorithm development

The general logic of the algorithms developed was that for each MoA there were two classes to which every experiment could belong to: either active or inactive. Then, an approach based on clusters was developed: for each MoA the mean and variance of all observations with $MoA = 1$ and $MoA = 0$ was computed, so each experiment can be compared against two clusters on each MoA. The second crucial step is to define a notion of distance, for which two methods were tried. First was to compute the euclidian distance, also known as norm-2, for which the square sum of the differences between the new experiment and the center of cluster with active (and inactive) MoA was computed; as a second metric, the Mahalanobis distance was used, which standardizes said sum of squares by the inverse of the covariance matrix. It is worth mentioning that the high number of covariates resulted in singular covariance matrices and thus, a pseudo inverse built from the

k main components in the singular value decomposition was used instead, where k was set to select eigenvalues greater than 10^{-5}.

The next challenge was to translate notions of distance into probabilities. This was done with an intermediate step computing weights of class belonging, which are later used to compute probabilities, but do not add to 1. The following weighting functions were used.

$$\begin{aligned} inverse(d, power) &= (1/d^{power}) \\ attraction(d, mass) &= (mass/d^2) \\ softmax(d, power) &= exp(-power * d) \end{aligned}$$

Where d is the distance to cluster center, mass is the number of observations in cluster and power is a parameter of choice. Note here that higher powers will differentiate more the differences in distances, skewing the distributions more to the extreme values, a value of 1 and 5 were used in both functions resulting in 5 ways of computing weights. Finally, probability of MoA being active was defined as:

$$P(MoA = 1) = \frac{w1 * n1}{w1 * n1 + w0 * n0} \tag{2}$$

Where $w1$ is the weight related to cluster $MoA = 1$ and $n1$ is the number of observations used to build the cluster. That is, weights (from same weight function) are turned into probabilities by their weighted average.

3 Results

Using a 2-fold cross-validation scheme, the best performing algorithm was inverse distance weights, with a log loss score of 0.142, which represents worse performance than the mean, a result from the rareness of outcomes (covariates can have more noise than signal). It is also worth noticing that algorithms based on the Mahalanobis distance, which encode more information on the data, performed worse than the euclidian distance, which is worth to investigate, and tune parameters and scaling more finely.

Figure 2: Risk by algorithm and distance metric

Algorithm	Euclidian	Mahalanobis
Inverse (Power=1)	0.142	0.4
Inverse (Power=5)	2.95	3.28
Attraction	2.95	16.3
Softmax (Power=1)	2.06	15.2
Softmax (Power=5)	0.69	14.8