mini project3

Xin Hao Jian Xun Yu Jinxing
Department of Computer Science

Abstract

This is a short report for the final-project of the class Math6380 in HKUST. The project we choose is Drug Sensitivity Ranking.
The invitation link at kaggle is https://kaggle.com/join/math6380project3

1 introduction

We choosed Drug Sensitivity Ranking and participated in the kaggle inclass contest. We are the team "3654". Our best submission gets accuracy of **94.74%** The dataset provides 265 Drugs with experiments on 990 cancer cell lines. Drug sensitivity is measured by IC50. There are 3 million parwise sensitivity comparisons for different durgs on each cells, and 1250 demensional gene describtion of eath cell line. For example, in cell1, the drug1 may be more sensitive than drug2, then the pair <drug1,drug2,cell1> will be 1. Among the 3M samples, 1/5 (.6M) comparison values are missing. Our job is to predict these values based on the information given.
We formulate the drug sensitivity pairwise ranking as a classification problem and present a classification model based on HodgeRank theory, to predict whether a durg is more, less, or equal sensitive than another drug on a cellline. We further extend the model by adding parameters to adaptively weight the binary gene feature vector, achieving better prediction accuracy on experiments.

2 Drug Sensitivity Rank Models

In this section, we introduce our drug sensitivity rank models based on the standard HodgeRank score. Different from the vanilla HodgeRank score, our model further incorporates the binary features of celllines and uses a softmax function to predict the probabilities of different pairwise comparison values. We formulate the parwise rank sensitivity prediction as a classification problem and use the cross-entropy between the predict probability and the ground truth pairwise rank label as the training objective function. The objective function is optimized via SGD (Stochastic gradient decent).
Let there be M genes, N celllines, and K drugs. Let $X(k) \in \mathcal{R}^M$ be the binary genetic feature vector of cell line k, $\beta_1(i) \in \mathcal{R}^M$ be the vector representation of drug i, $\beta_0(i) \in \mathcal{R}$ be the inital score of drug i, where $\beta_1 \in \mathcal{R}^{K\times M}$ and $\beta_0 \in \mathcal{R}^K$ are model parameters.
Given a cell line k with genetic feature vector $X(k)$ and pairwise sensitivity of drug i and j on k, $y(k,i,j) \in \{-1,0,1\}$, the rank score is computed as

$$s = \beta_0(i) - \beta_0(j) + X(K)^T(\beta_1(i) - \beta_1(j)). \tag{1}$$

The score is transformed to prediction probability by

$$p = softmax(Ws + b), \tag{2}$$

where $p \in \mathcal{R}^3$, $p(0), p(1), p(2)$ indicate the probability that the parwise rank value is -1,0, or 1 respectively, $W, b \in \mathcal{R}^3$ are model parameters, $softmax(x)_j = \frac{e^{x_j}}{\sum_{i=1}^K x_i}$.The overall loss is the negative log-likelihoods of the rank prediction:

$$\mathcal{L} = -\log(p(y(k,i,j)+1)) \tag{3}$$

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Let $t = Ws + b$, $\hat{y}(k,i,j) \in \mathcal{R}^3$ be the one-hot encoding representation of ground truth label $y(k,i,j)$. The gradients of model parameters can be caculated by the chain rule:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial t} &= p - \hat{y}(k,i,j) \\
\frac{\partial \mathcal{L}}{\partial W} &= \frac{\partial \mathcal{L}}{\partial t} s \\
\frac{\partial \mathcal{L}}{\partial b} &= \frac{\partial \mathcal{L}}{\partial t} \\
\frac{\partial \mathcal{L}}{\partial s} &= W^T \frac{\partial \mathcal{L}}{\partial t} \\
\frac{\partial \mathcal{L}}{\partial \beta_0(i)} &= \frac{\partial \mathcal{L}}{\partial s} \\
\frac{\partial \mathcal{L}}{\partial \beta_0(j)} &= -\frac{\partial \mathcal{L}}{\partial s} \\
\frac{\partial \mathcal{L}}{\partial \beta_1(i)} &= X(k) \frac{\partial \mathcal{L}}{\partial s} \\
\frac{\partial \mathcal{L}}{\partial \beta_1(j)} &= -X(k) \frac{\partial \mathcal{L}}{\partial s}
\end{aligned}$$

Since the cell line feature $X(k)$ is a binary feature vector, we also propose an extention to our model by adding model parameters $w_1 \in \mathcal{R}^M$ to weight the cell line features. Then the rank score is modified as

$$s = \beta_0(i) - \beta_0(j) + (X(K) \odot w_1)^T (\beta_1(i) - \beta_1(j)), \tag{4}$$

where $\odot$ represents element-wise product of vectors.

3 Methodology

We tested both the original model (*Original*) and the model with extension of w_1 (*Extended*). For both models we used the same training settings, with epoch=1000, batch size=1000, learning rate=0.001, and initial weights following Gaussian distribution.

4 Results and Discussion

We compared two models in four aspects: training loss, training accuracy, validation accuracy, and test score from kaggle. The results are shown in Table 1.

model	training loss	training accuracy	validation accuracy	test score
Original	59.0390	0.9753	0.9317	0.9418
Extended	41.3801	0.9858	0.9372	0.9474

Table 1: Experiment results

From the results in Table 1, the *Extended* model performed better in all four aspects. We believe that this is because the w_1 parameter makes the model more flexible, so that it can cross some local optimal regions more easily. We also noticed that the *Extended* model had lower training loss than the *Original* model, while for other three aspects, especially for validation accuracy and test score, these two models had no big differences. We think this can be explained by the difference of model complexity. The *Extended* model is more complex than the *Original* model, so it can fit the training data better, but the accuracy is too high to improve much.

5 Remark on Contributions

The project is finished under the discussion and close collaboration of our group members. Jinxing Yu wrote the code skeleton and wrote the model part. Hao Xin and Xun Jian proposed different changes to the models. Hao Xin wrote the introduction part and Xun Jian wrote the discussion part.