UW DS350 Project Felix Huang

1. Data pre-processing : to prepare job description data
2. Naïve-Bayes for ‘associate-needed’ label 🡺 accuracy: 0.9208861
3. Naïve-Bayes for ‘bs-degree-needed’ label 🡺 accuracy: 0.8924051
4. Neural Network for 'associate-needed' label 🡺 accuracy: 0.943038
5. Neural Network for 'bs-degree-needed' label 🡺 accuracy: 0.914557
6. Logistic regression for 'associate-needed' label 🡺 accuracy:

0.9588608

1. Data pre-processing : to prepare job description data

Input data train.tsv was obtained from a HackerRank competition sponsored by Indeed.

Original data comes from Indeed. The input data has 1580 rows and 2 columns. Two

columns are job description text and tags (labels). There are 12 labels. In this project, we

try to do machine learning and predict two labels about degree required:

'bs-degree-needed' and ‘associate-needed’. According to Indeed rule, these two labels

are mutually exclusive. For example, a job description contains “qualification: Associates

degree or Bachelors degree”, and then the label would be ‘associate-needed’ only (the

lower level degree is required).

1. Naïve-Bayes for ‘associate-needed’ label

Posterior = Likelihood \* (Prior / P(data | parameters))

Previously I tried to use original text after cleaning and do Naïve Bayes, but R crashed around Bayes train and prediction (Line 154,155) because there are too many columns in text\_term\_matrix.

Then I began to use key words and expand from these key words to include neighborhood context by function expand\_context(), to reduce number of terms in text\_term\_matrix. For text\_term\_matrix, words with low frequency are not removed because some key words with low frequency in a job description may be important for prediction. (e.g., Associates Degree)

This time Naïve Bayes can run through. Based on results of first a few runs, I adjusted key words and context expansion. This leads to accuracy 92.1%.

1. Naïve-Bayes for ‘bs-degree-needed’ label

According to Indeed rule, ‘bs-degree-needed’ and ‘associate-needed’ labels are mutually exclusive. For example, “Associates degree is required and Bachelors degree is preferred.” is in input text and then the degree label is ‘associate-needed’ only (no bs label).

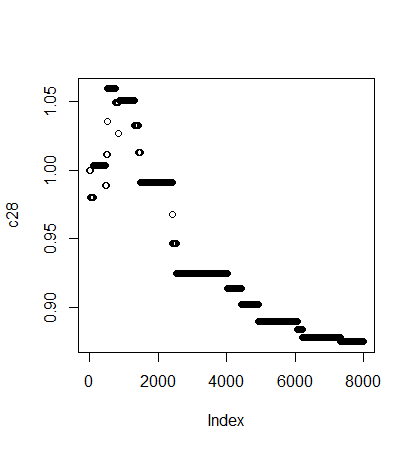
Previously I tried to use both bs and as key words and hope Bayes algorithm can figure out positive and negative logic. But Bayes in R seems to be weak about negative logic (e.g., Associates degree leads to no bs-degree-needed label). Then I used only bs key words. When we do prediction on test set after train by Bayes with bs key words only, remove (as from (1) == TRUE and bs == TRUE) from (bs == TRUE). This leads to accuracy 89.2% for ‘bs-degree-needed’ by Bayes.

1. Neural Network for 'associate-needed' label

This uses similar approach as in neural\_nets.R (L323-541) in our class. For each loop in train, we pick a random row, calculate inner product of coefficients (vars in nn) and input data and network\_out, and then use that to determine pull(sign) to calculate gradients and update coefficients. Learning rate (step\_size) decreases from 0.1 to 0.001 during train process. For every half loops, step\_size is divided by 2. By this learning rate control, we try to move effectively in early stage and help convergence in late stage in neural network train. On test set, this leads to average accuracy 94.3% for 'associate-needed' label.

Each round (8000 loops) in train takes about 5 min to run. In (3), 5 rounds are done to get average accuracy 94.3%.

Coefficients of an important feature ‘associatesDegree’ during train (8000 loops) is plotted as follows.

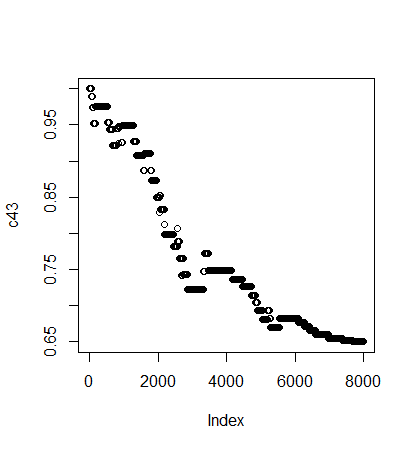


1. Neural Network for 'bs-degree-needed' label

This uses similar neural network approach in (3). This time I include as key words and try to see whether it can do better than (2) Bayes on negative logic (job descriptions with stronger ‘as’ lead to no ‘bs’ label). Neural network (4) seems to do better on negative logic. When I tried to improve it more, I added simulated annealing to train. For previous pull=0, coefficients can randomly move by a probability upper bound 0.26 and this probability upper bound decreases to 0 as train process goes on. Roughly we accept necessary moves (pull=1, pull=-1) and allow small portion for random moves from (pull=0). Simulated annealing increases chance to climb some hills and helps improve accuracy from 86% to 91.4%. No intersection removal of ‘bs’ and ‘as’ as in (2) is done in (4). Neural network figures out positive logic (e.g., only “Bachelors degree” appears in text) and negative logic (no ‘bs’ label, e.g., Associates degree exists) by itself.

Each round (8000 loops) in train takes about 5 min to run. In (4), 3 rounds are done to get average accuracy 91.4%.

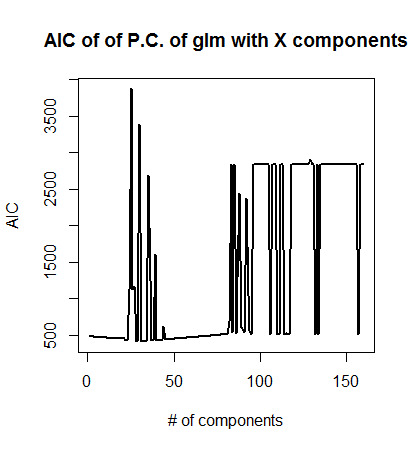
Coefficients of an important feature ‘bachelorsDegree’ during train (8000 loops) is plotted as follows.



1. Logistic regression for 'associate-needed' label

This follows data preparation work in (1) and (3) for 'associate-needed'. All features turn to numeric. For glm in (5), in previous run I got accuracy 86%. Then I began to reduce key words and context expansion (7 -> 2) from key words, the accuracy is improved from 86% to 95.9%. (5) uses less ‘as’ key words and smaller context expansion than (1) and (3).

Principal components are calculated. (# components) vs AIC is plotted as follows.



28 principal components lead to minimum AIC 409.6226.