Feature Selection for Ranking
By: Xiubo Geng, Tie-Yan Liu, Tao Qin, Hang Li

Siddharth Baronia

Background and Relation -

This paper presents a potential solution to the existing problem of ranking documents in information retrieval area and how critical feature selection is in ranking related problems. Ranking is the most crucial aspect of information retrieval as it relates to the relevancy of the retrieved information. So far most of the methods we studied for classification for feature selection can also be applied to ranking but a new selection method is proposed in this paper. The new method of feature selection is based on Greedy Search Algorithm which takes into account similarity between two features and importance score of individual feature. The goal is to maximize the total importance score and minimize total similarity score so similar features can be avoided. The interest behind selecting this paper is to understand this new method of feature selection and its application in ranking and applying it on a classification problem covered in class and compare the results of model fitting, training and testing errors to other methods we covered.

Ranking for a given set of objects is a score given to each of them by following a criteria and then sort them according to their scores. All of the objects in the set are expressed by number of features and the size of these features can be large with most of the features not even effecting ranking. Throughout this coursework the importance of choosing the right features and right number of features significantly impacted the results of modelling and error for different methods and different modelling techniques yielded different set of features as important. I would like to explore this method of feature selection by applying it directly to Wheat data for

classification problem and to find what features are the most important in providing the closes prediction and least training, testing errors.

Through classification we try to solve the problem of predicting categorical responses or classes and explanatory variables are the features that define the class. Standard loss function is defined and misclassification rate on training, testing and validation set. The general method of classification is to find the boundary function or decision boundary to classify classes. The methods covered in class are Linear Discriminant Analysis, Quadratic Discriminant Analysis, Logistic Regression, Naïve Bayes, Smooth Binomial, Tree based, Support Vector Machine and these methods take some tuning parameters for model fitting using training set and using that model prediction is made on testing set and errors can be calculated [1]. Using the same techniques, we covered so far, for the purpose of this paper I have used Wheat data and used Greedy Search Algorithm to find the right set of features that can provide the best prediction and least error.

Features of Paper -

This paper states that development of supervised learning algorithms such as Ranking SVM and RankNet allow us to incorporate more features in ranking models and since having more features can lead to unnecessary over-fitting, appropriate feature selection has become even more important. On other hand feature selection also improves efficiency of training and when large number of features are present running a model on them can be costly and time consuming. For classification purpose there are three different types of feature selection methods – filter, wrapper and embedded. The paper focusses on filter, in which feature selection is a preprocessing step

and can be independent from learning. A score is calculated for each feature and then top scoring features are selected.

Ranking and classification are different in ways such that evaluation measures in ranking are more based on precision and that in classification is based on precision and recall. Moreover, in ranking ordered categories are used while in classification categories are flat. The new proposed method uses loss function to measure important features, compares similarities between features to remove redundant and used greedy search algorithm to optimize selection problem.

Feature Selection Method -

Goal here is to select t out of m features where $1 \leq t \leq m$ and feature set is $\{v_1, v_2, v_3 \ldots v_m\}$. To formulate feature selection method, we need importance score and similarity score.

Importance score of each feature and similarity between features are calculated. To calculate importance score MAP (Mean Average Precision), NDCG (Normalized Discounted Cumulative Gain) or loss function can be used. MAP measures the precision of ranking results where precision is the occurrence of true positive in all the predicted positives. The average precision is calculated as –

$$AP = \sum_{n=1}^{N} \frac{P(n) \times pos(n)}{\text{number of positive instances}}$$

where P(n) is precision at n measures, pos(n) is binary function indicating whether the document at positive n is positive, n is position, N is total objects.

NDCG, measures ranking accuracies when there are multiple levels of relevance. For a query NDCG at position n is –

$$N(n) = Z_n \sum_{j=1}^{n} \frac{2^{R(j)}-1}{\log(1+j)}$$

where n is position, R(j) is score of rank j, Z_n is normalization factor that equate NDCG at n to 1. MAP and NDCG are used to compute importance scores of features. For my implementation during classification of Wheat data I have used MAP as a measure which can be calculated using **mapk** function present in **Metrics [2]** package in R.

Second is the similarity between features which tells about correlation between two features and helps in removing the redundancy. So for the purpose of finding similarity we regard each feature as ranking model and find similarity using one of the measures such as Spearman's footrule, rank correlation, Kendall's tau. For this paper Kendall's tau is used which is –

$$\tau_q(v_i, v_j) = \frac{\#\{(d_s,d_t)\in D_q \mid d_s \prec_{v_i} d_t \text{ and } d_s \prec_{v_j} d_t\}}{\#\{(d_s,d_t)\in D_q\}}$$

D_q is set of instance pairs (d_s,d_t) in response with respect to query q, #{.} represents the number of elements in a set, $d_s \prec_{v_i} d_t$ implies that instance d_t is ranked ahead of instance d_s by feature v_i.

For my implementation during classification I have used the function **Kendall** available in **Kendall [3]** package of R.

As mentioned before we need to select features with largest importance score and smallest similarity scores, hence we will try to maximize the difference between these two measures -

$$\max \ \sum_i \omega_i x_i \ - \ c \sum_i \sum_{j \neq i} e_{i,j} x_i x_j$$
$$s.t. \ \ x_i \epsilon \{0,1\} \ \ i = 1, \dots, m$$
$$\sum_i x_i = t$$

Where w_i is importance score of feature v_i and $e_{i,j}$ denotes the similarity (Kendall's tau) between feature v_i and feature v_j. t is number of selected features and x_i is 1 or 0 depending on if a feature is selected or not.

To optimize the formula above a Greedy Search Algorithm (GAS) is used for feature selection, which is showed in Figure 1 along with the R code that I presented for the same.

Algorithm GAS (Greedy search Algorithm of feature Selection)

1. Construct an undirected graph G_0, in which each node represents a feature, the weight of node v_i is ω_i and the weight of an edge between node v_i and node v_j is $e_{i,j}$.
2. Construct a set S to contain the selected features. Initially $S_0 = \emptyset$.
3. For $i = 1 \ldots t$,
   (1) Select the node with the largest weight, without loss of generality, suppose that the selected node is v_{k_i}.
   (2) A punishment is conducted on all the other nodes according to their similarities with v_{k_i}. That is, the weights of all the other nodes are updated as follows.
   $$\omega_j \leftarrow \omega_j - e_{k_i,j} * 2c, \quad j \neq k_i$$
   (3) Add v_{k_i} to the set S and remove it from graph G together with all the edges connected to it:
   $$S_{i+1} = S_i \cup \{v_{k_i}\}, \; G_{i+1} = G_i \backslash \{v_{k_i}\}$$
4. Output S_t.

```
final.features <- c()
final.weight <- c()
temp.prec.matrix <- precision.matrix

for (r in 1:6) {
  max.map.feat <- temp.prec.matrix[which.max(temp.prec.matrix[,2]),1]
  final.features <- c(final.features,max.map.feat)
  final.weight <- c(final.weight,max(temp.prec.matrix[,2]))
  for (t in 1:nrow(temp.prec.matrix)) {
    for (h in 1:nrow(feat.kendall.df)) {
      if ((feat.kendall.df[h,1] == temp.prec.matrix[t,1] && feat.kendall.df[h,2] == max.map.feat) ||
          (feat.kendall.df[h,2] == temp.prec.matrix[t,1] && feat.kendall.df[h,1] == max.map.feat)) {
        temp.prec.matrix[t,2] <- as.numeric(temp.prec.matrix[t,2]) - 2*1*as.numeric(feat.kendall.df[h,3])
      }
    }
  }
  temp.prec.matrix <- temp.prec.matrix[which(temp.prec.matrix$feature != max.map.feat),]
  print(temp.prec.matrix)
}
```

Figure 1 - GAS Procedure for feature selection and R code

For the procedure on left S_t contains the features in decreasing order of importance and on the right hand code side *final.features* contains the list of of features in decreasing order of importance for Wheat data.

For the purpose of this paper and topic two ranking models – Ranking SVM and RankNet are used. RankNet employs a neural network as ranking function and relative entropy as loss function. It also employs gradient decent to minimize the total loss with respect to the training data and validation set to avoid use of local optimum. Ranking SVM is an extension of SVM which utilizes instance pairs and their preference labels in training. Feature selection is run on training set, on selected feature a ranking model is trained and tuning is done using a validation set. Finally, with the obtained model ranking is done on test set.

To show a working implementation I will keep the scope of this paper to the point of understanding how feature selection is done through GAS and how to use MAP and Kendall's Tau for the purpose of calculating importance score of features and similarity scores. Instead of

ranking documents my implementation using these calculations will target better classification to align with the class content and other assignments.

Classification of Wheat data -

Using the methods of Greedy Search Algorithm by using MAP and Kendall's similarity measure I have implemented this feature selection technique for classification on Wheat data. About the data - it has 6 features (class, density, hardness, size, weight, moisture) and one class variable type which has values of Healthy, Scab and Sprout. For the purpose of my implementation I changed the class variable to hold only two types – Healthy and Unhealthy. The variable class is converted to factor – 1 (srw) and 2 (hrw). The entire dataset is divided into training and testing set. For every pair for features, in total 15 combinations, I calculated the similarity by using **Kendall** function.

Usage: [Kendall - **Kendall (x, y)**, x and y are variables for which similarity is needed]

Now for every feature we will tune an SVM model on training data using ***tune.svm*** [4] (e1071 package) against "type", with *gamma* ranging from [10^{-2} to 10^{-1}] and *cost* ranging from [10^{2} to 10^{3}], cross = 10 and kernel type used is *radial*. Best gamma and cost is extracted from this tuning and then an ***svm*** model is fit using training data and these values from which we predict using training data. This whole procedure is repeated for all features individual fit. With this I have all the predictions (Healthy/Unhealthy) from individual feature.

Now for every feature a weight is calculated which is measure of importance by using Mean Average Precision (MAP). The functions used in **mapk** from **Metrics** package –

Usage: [MAP – mapk (k, actual, predicted), k is length of predicted sequence]

Now with importance score and similarity score available an undirected graph is constructed in which every feature is connected to other. Right side of Figure 1 is the code for greedy search that is written to extract features in decreasing order of importance based on their weight. Through out the process of GAS we decrease the size of graph by removing the feature with highest weight and readjust the weights of remaining features. Through my implementation the order of features I found is -

	final.features	final.weight
1	density	0.815
2	moisture	-21.36
3	size	-43.36
4	classnum	-45.36
5	hardness	-57.36
6	weight	-63.36

So the feature importance in decreasing order is – ***(density, moisture, size, classnum, hardness, weight).***

Now with these features in order I tuned a SVM model on training data with same gamma and cost range, cval = 10 with decreasing features to find the right combination of features that gives smallest training error. Once best gamm and cost is extracted an svm model is fit and training error is calculated. Following is the training error per combination of features –

```
(density+moisture+size+classnum+hardness+weight) =  0.16
(density+moisture+size+classnum+hardness) = 0.16
(density+moisture+size+classnum) = 0.13  <<<<<<<<< best
(density+moisture+size) = 0.1942857
(density+moisture) = 0.1828571
(density) = 0.1942857
```

From this table its evident that fitting model with 4 features – ***density, moisture, size, classnum*** gives the least training error. The model is made with these features and training set and then test data set is used to make test prediction and calculating testing error.

```
> # training error
> wheat.svm.train.error.2
[1] 0.13
>
> # testing error
> wheat.svm.test.error.2
[1] 0.2666667
```

Training confusion matrix

```
             predicted_tr
observed_tr Healthy Unhealthy
  Healthy        61        11
  Unhealthy      15       113
```

Testing Confusion matrix

```
             predicted_te
observed_te Healthy Unhealthy
  Healthy        17         7
  Unhealthy      13        38
```

About Code -

A complete working R code (sbaronia-project2.R) is written using Wheat data and SVM methods. The code is well commented in parts to understand working of different sections. The code is emailed separately to professor.

Comparison with other methods -

Method	**Training Error**	**Testing Error**
LDA	0.30	0.293
QDA	0.255	0.253
Multinom Logistic Regression	0.285	0.28
Logistic Regression	0.19	0.24
Smooth Binomial	0.79	0.98
GAM	0.845	0.80
Naïve Bayes Gaussian	0.26	0.29
Naïve Bayes Kernel	0.29	0.31
Tree	N/A	0.38
SVM Radial	0.24	0.28
SVM Polynomial	0.205	0.293
Gradient Boosting	0.255	0.333
GAS – Paper	**0.13**	**0.266**

It is very evident from the table above that this new method gives the best training error for given set of training data which is consistent in all methods and the test error is highly competitive in comparison to most of the methods, performing very close to logistic regression and QDA.

Conclusion -

Since there are many methods that can interchangeably be used for classification and ranking and feature selection being the most important criterion in this regards, I have used the new method of feature selection for ranking proposed in this paper by using GAS, MAP and Kendall's tau for classification to see if the features selected through this method comes close to what I did in class assignments. Although ranking and classification are different the method worked fine for classification. Maximizing total importance and minimizing similarity score is the gist of the method which is done using famous Greedy Search Algorithm. The method worked great and reported training error of 0.13 and testing error of 0.266. So coming up with right set of features and then fitting the model can be a way to modelling and classification.

Reference:

[0] Paper - http://research.microsoft.com/en-us/um/people/taoqin/papers/qin-sigir07b.pdf

[1] Lect 16 - https://canvas.sfu.ca/courses/28781/pages/lecture-16-intro-to-classification?module_item_id=575849

[2] Metrics package - ftp://cran.r-project.org/pub/R/web/packages/Metrics/Metrics.pdf

[3] Kendall package - https://cran.r-project.org/web/packages/Kendall/Kendall.pdf

[4] E1071 package - https://cran.r-project.org/web/packages/e1071/e1071.pdf