Journal of
Applied & Computational Mathematics

Tesiero, J Appl Computat Math 2016, 5:4
DOI: 10.4172/2168-9679.1000315

Short communication Open Access

Discovery of Long Tail Keywords in Paid Search

Tesiero J*

Principal Data Scientist Consultant, University of Maine, USA

Abstract

The following work describes an elegant, efficient keyword clustering method to discover long tail keywords in paid search data. In keyword auctions, such words often go undiscovered as their cost in being bid to higher ranking positions is deemed too high to justify the potential of significantly added conversion revenue. By discovering clusters with low volume keywords and established, high-performing and high volume keywords, the quality of the low volume (long tail) keywords is inferred by association.

After a brief introduction, the data used to train the clustering algorithm is described. Then, the data reduction process (the discovery of the most predictive features) is described. We then describe the method, followed by the results and interpretation.

Keywords: Clustering method; Marin data; Matrix yield; Nonlinear

Introduction

The keyword clustering model described in this document will be used to target keywords that are most likely to convert during a given time period, typically on a particular day. Its utility stems from being able to identify certain keywords that have a low number of impressions per day as potentially convertible and revenue-generating, but only if placed in a higher position where they would receive more impressions, and in producing a cluster map that separates clusters of high conversion potential keywords from clusters of low conversion potential. This makes it easier to determine which keywords campaigns should be constructed [1].

Data

Description of the data sets

The datasets used to build the keyword clustering model, as well as to validate it, were drawn from the Marin data. The Marin data consists of a set of Excel files, segmented by month, resulting from the application of the Marin software application on a database containing all of the keyword queries performed for each month.

The Marin data used specifically for the building and validation of the keyword clustering model spans the time period from November 2012 to July 2013. There are approximately 100,000 keywords for each month of data, but they are not unique. The unique rows in the Marin data are determined by the combination of date, keyword, publisher, campaign, and group. The rows in the Marin data correspond to daily values across all of the features in the data sets.

Description of features

There are a total of 11 features in the Marin data set that could conceivably be used for modeling. The original 11 features in the Marin data are the following: Quality Score, Impressions, Clicks, CTR, Avg. Position, Publication Cost, Avg. CPC, Cost Per Impression, Search Bid, Avg. Bid, and Headroom. These features can be grouped into three individual factor groups: Volume, Quality, and Cost. For example, the volume factor group would contain features such as Impressions and Clicks, for the Quality group, Quality Score and CTR, and for cost it would be CPC and Cost per Impression [2].

Variable Selection for the Model

Method

A multivariate Pearson correlation analysis was performed with the 11 features and the response variable, the cost per conversion. The Pearson correlation function is given by the following equation:

$$\rho(X,Y)=\sum_{i=1}^{N}\left(\left(X(i)-\mu(X)\right)*\left(Y(i)-\mu(y)\right)\right)/\left(\sigma(X)*\sigma(Y)\right)$$

where X is a feature, Y is the response variable, and {μ(X), σ(X)}, { μ(y),σ(Y)} are the mean and standard deviation of the feature distribution X and the response variable distribution Y respectively. The above equation represents one element of the 12 × 12 correlation matrix, which is then diagonalized. The diagonalization yields the amount of variance explained by each feature X with the response variable Y, removing the pair-wise feature-feature correlations. The normalized eigenvalues of the diagonalized matrix yield the variance percentage explained by a given feature, so the dimensionality of the system can be reduced by keeping only the features associated with the largest eigenvalues [3].

The original 11 features in the Marin data are the following: Quality Score, Impressions, Clicks, CTR, Avg. Position, Publication Cost, Avg. CPC, Cost Per Impression, Search Bid, Avg. Bid, and Headroom.

Results and interpretation

After applying the above analysis, the reduced feature space yields three variables: namely Impressions, CTR, and Avg. CPC. Each of these three remaining features has the elegant quality that it represents a particular factor group; Impressions (Volume), CTR (Quality), and Avg. CPC (Cost), which allows a clear interpretation of a keyword in terms of the relative amount that those factors contribute to conversion prediction for each keyword, once the logistic regression is applied. Also, being three-dimensional lends the keyword representations

***Corresponding author:** Tesiero J, Principal Data Scientist Consultant, University of Maine, USA, Tel: 207 581 1865; E-mail: www.blipiq.com

Received April 22, 2015; **Accepted** July 25, 2016; **Published** July 29, 2016

Citation: Tesiero J (2016) Discovery of Long Tail Keywords in Paid Search. J Appl Computat Math 5: 315. doi: 10.4172/2168-9679.1000315

to better visualization, as the whole feature space can be visualized without projections. One can get a visual understanding of how well conversion rates can be predicted by the cluster map, which shows the keyword clusters as circles centered about the cluster centroids and gives an intuitive picture of cluster compactness and separation, which are integral to the generalization of parameters derived from the clusters to validation data sets.

Standardization of variables

As a pre-processing step to clustering, the raw values of the features representing the keyword data are standardized across all keywords. The standardized values of the features are related to the raw values, through the following equation:

$$z(i,f)=\frac{\left(X(i,f)-\mu(X,f)\right)}{\sigma(X,f)}$$

where X(I,f) is the raw value of the f^{th} feature for the i^{th} keyword, μ(X,f) is the mean value of X for feature f, σ(X,f) is the standard deviation of X for feature f, and z(I,f) is the z-score of the ith keyword for the f^{th} feature. This transformation is necessary to put the data on the same scale, since it will be used to perform clustering and regression, which are both sensitive to the relative scale of the features of the data, since they are based on distances in the feature space.

Clustering

Motivation

The motivation for clustering the feature space comes from two sources; first, the business needs to find keywords that achieve low volume but are nevertheless high quality queries. This comes from association in the same keyword cluster with high volume queries that are known to be strong (in the sense that they have a high Quality Score as calculated by Google). The second source is the practical reality that regression, even a robust regression such as logistic regression, fails to produce good conversion rate predictions on a daily level without clustering.

Method

The clustering method used for keyword clustering of the Marin data was k-means, with a Euclidean distance metric. K-means falls within the general class of global optimization algorithms, and its global minimum ideally results in the largest intra-cluster to inter-cluster similarity ratio. When used as a supervised learning method, the global optimum also increases generalization in predictive accuracy between the training set and the validation set. In practice, the global minimum is never reached in high dimensions, but can be achieved in the effectively three dimensional problem here.

The K-means algorithm works in the following way:

- The value of K (the number of clusters) is selected or determined by some procedure.
- A sampling technique is applied to "seed" the dataset with K of the N points in the dataset, which represent the initial centroids from which the initial set of clusters shall be determined.
- The distance from a point to each centroid is calculated, the point is then assigned to the cluster of the closest center. Repeat for all points.
- Calculate the new centroids based on the above calculations.
- Repeat steps above until convergence.

The value of K that was used in this first version of keyword clustering was 100. This value was determined by using a range of K from 90 to 110 and finding the value of K that yielded the highest predictive accuracy in the validation set. In the document Optimal_K_Algorithm.pdf, a rigorous way of calculating the optimal K for any given data set will be described (this is the approach for determining K in the next version of keyword clustering). This method is a closed form equation for optimal K, subsequently making it easy to implement. Furthermore, it is dynamic and therefore should scale well.

The K-means algorithm is implemented in MATLAB. There are a range of input options for sampling and distance. The sampling technique used for obtaining initial cluster centroids was uniform sampling, in order to minimize the probability of bias toward any particular feature or time period of the data, and the distance metric used is Euclidean distance.

Cluster evaluation

The clusters are evaluated based on the ratio of the intra-cluster and inter-cluster densities. The intra-cluster, or within cluster density, is a measure of how densely packed the points are around the centroid of the cluster. The inter-cluster density is given by the relative separation of the cluster centroids. Both of these quantities are outputs of the MATLAB implementation of K-means. The intra- cluster density is given by sumd output argument in MATLAB, which is a K ×1 array containing the sum of the distances to the k^{th} cluster for all of the data points in the k^{th} cluster. The inter-cluster density is derived from the elements of the D matrix, which is an nxk matrix containing the distances from every point in the data set to each centroid:

$$D=\begin{matrix} D(1,1) & D(1,2)\ldots & D(1,100) \\ D(2,1) & D(2,2)\ldots & D(2,100) \\ D(N,1) & D(N,2)\ldots & D(N,100) \end{matrix}$$

The output of the K-means algorithm also contains the data points that are members of each cluster, in the variable IDX. Therefore, by defining the intra-cluster to inter-cluster ratio as Ω, we have the following equation:

$$\Omega=\sum_{j}\left(\sum_{\forall idx(i)=j} d\left(i,c(j)\right)/\left(\sum_{i} D(i,j)-\sum_{\forall idx(i)=j} d\left(i,c(j)\right)\right)\right)$$

where j indexes the clusters, and i indexes the data points, $\forall idx(i)=j$ means all of the data points that are members of the j^{th} cluster, c(j) is the centroid of the j^{th} cluster, and d(i,c(j)) is the distance between the i^{th} point and the j^{th} cluster. The quantity Ω gives a global picture of the intra-cluster to inter-cluster density, without examining variations within specific regions of the cluster map. This quantity was calculated for the ranges of K used in determining the best initial K to use, and it is not surprising that the above quantity is minimized at the same value of K, where the best validation results were achieved, K=100.

The above Figure 1 is a cluster map in the case K=100 (100 clusters) of 123,000 data points in the training set used for deriving parameters which are then used in the prediction of daily conversion rates on keywords in the validation set. The circles in the figure above are centered on the cluster centroids, with the radii of the circles representing the average standard deviation over each feature. As can be seen by visual inspection, there is minimal overlap, which indicates that the features used are good predictor candidates for the response variable and the cost per conversion, from which the probability of conversion can be derived.

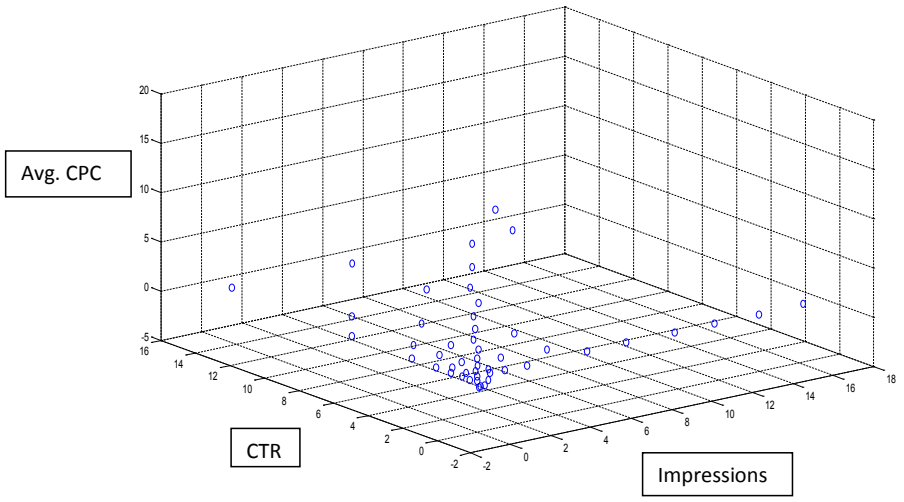

Figure 1: Cluster map.

Impressions	CTR	Avg. CPC		
21.0066256	40.56581458	0.519442975	2.582877364	38.77371228
29.34451951	26.47780603	7.53223E-09	90.73928485	525.0433617
330.129338	831.722181	4.92006E-08	12.30959908	1.596419666
14.66207397	42.10329333	1.998791011	2.246890541	1.998794857
16	18	21	30	32

Table 1: The weights from the logistic regression.

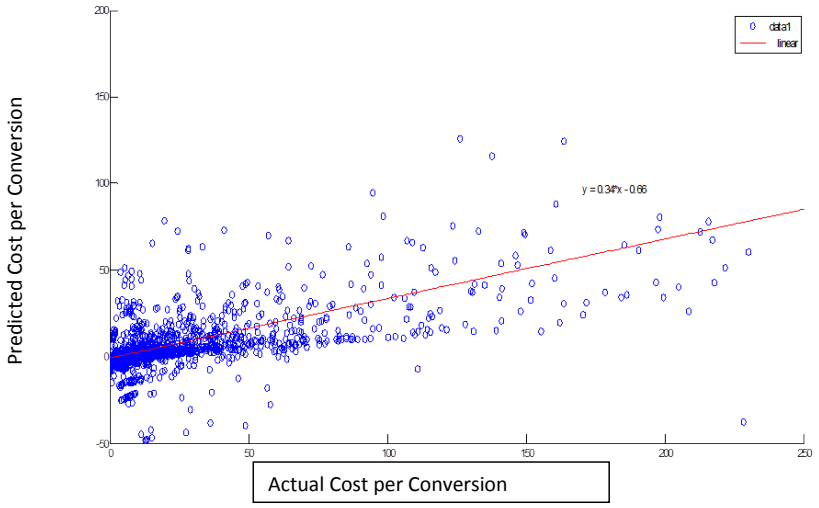

Figure 2: Graph of predicted cost per conversion vs. actual cost per conversion.

Logistic regression

Logistic regression is a type of regression used when the response variable is a discrete choice variable with a finite number of values, and also when the relationship between the feature variables and the response variable is nonlinear, in particular sigmoidal.

The sigmoid function f(X) is given by:

$$f(X)=\frac{1}{1+\exp(-X)}$$

where $X=w_0+(w_1*x_1)+(w_2*x_2)+\ldots+(w_n*x_n)$, and the $\{x_1,\ldots,x_n\}$ are the feature variables in the regression, $\{w_0,\ldots,w_n\}$ are the weights of the features determined by the regression, and w_0 is the bias term. f(X) is the response variable. In the case of the keyword clustering model developed here, n=3 (since we have a three dimensional problem, and f(X) is the normalized cost per conversion, which lies between 0 and 1 as the response function of a logistic regression should. Therefore, x_1, x_2, and x_3 correspond to the impressions, the CTR, and the average CPC respectively.

The logistic regression is applied separately to the clusters, as there is not significant lift above random prediction without bounding the regressions to continuous subsets of the feature space. With the clustering, conversion prediction based on a threshold function applied to the predicted cost per conversion results in 70% conversion prediction accuracy across 180,000 keywords.

The weights from the logistic regression are shown on some of the clusters below. The clusters are in the columns of the data. The first row is the bias term for each cluster; the next three rows are the respective weights for the impressions, CTR, and Avg. CPC for each cluster, while the last row identifies the cluster Table 1.

The five clusters in the table are of three different types. Clusters 16 and 18 have high weights for the CTR variable, so these tend to detect high quality keywords based on the CTR, with little variance in the impression volume or average CPC. The second type contains either poor quality keywords since the weights for all of the predictors are low, or good keywords only in a very small range of impressions, CTR, and average CPC. The third type has dominating impression volume weights, which indicates that the greatest variance lies in the impression volume. It is in this case that low impression volume keywords can be associated with high quality, high impression volume keywords, which suggests that these keywords would get more impressions and possibly more conversions, but only if ranked higher.

Results

The Figure 2 below is a scatter graph of predicted cost per conversion vs. actual cost per conversion. The strong correlation here suggests an ability to predict conversion rate, which is related closely to cost per conversion; significantly better than random prediction. The figure below is for 10,000 randomly selected keywords from the 180,000 keyword validation set, and the prediction accuracy in terms of predicting conversion rate is 70%.

References

1. Basu C, Hirsh H, Cohen W (1998) Recommendation as classification: Using social and content based information in recommendation. AAAI.
2. Goldenfeld N (1992) Lectures on phase transitions and the renormalization group. Addison-Wesley Publishing Company.
3. Guadawardana A, Meek C (2009) A unified approach to building hybrid recommendation systems.