Conversion Prediction for Advertisement Recommendation using Expectation Maximization

Sejal D
Department of Computer Science and Engineering
B N M Institute of Technology,
Bangalore
Email: sej_nim@yahoo.co.in

Shradha G
Venugopal K R
Department of Computer Science and Engineering
University Visvesvaraya College of Engineering,
Bangalore University,
Bangalore-560001

S S Iyengar
Florida International University, USA

L M Patnaik
INSA Senior Scientist
National Institute of Advanced Studies
IISc Campus, Bangalore

***Abstract*—Advertiser has to understand the purchase requirement of the users who are looking for a particular service to recommend advertisement. Once the users' demand is identified, advertisers can target those users with appropriate query. In this paper, predicting conversion in advertising using expectation maximization [PCAEM] model is proposed to provide influence of their advertising campaigns to the advertisers by understanding hidden topics in search terms with respect to the time period. Query terms present in search log are used to construct vocabulary. Expectation Maximization technique is used to learn hidden topics from the vocabulary. Least Absolute Shrinkage and Selection Operator (LASSO) is used to predict total number of conversion. Experiment results show that PCAEM model outperforms TopicMachine model by reducing Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for prediction.**

***Keywords*—Advertisement recommendation, Conversion Prediction, Expectation Maximization**

I. Introduction

Internet provides information to users; it takes input as a query and gives related results. Many service providers use this characteristics of the Internet to their advantage. Consider a query *create an android app for generating recipe* is entered by a user to the search engine. The user may be looking for online tutorials or classes for learning android programming. It is assumed that such a user is willing to pay for a service that can help him to learn Android program. The service provider that provides such services wants his advertisement to appear as the most relevant result for this query.

The query given to a search engine is called as the search term and resulting advertisements are called as keyword based advertising. The search engine market (SEM) agencies connect the two sides of this same coin, i.e., the customers and the service providers. The SEM agencies maintain a large collection of keywords for their clients, the service providers. As the services provided by these clients grow, the number of keywords also increase and need to be managed. SEM agencies conduct experiments on the keywords to test their weight, relevance and usage. These tests help the clients to select the best variant for the keywords related to the services they provide. However, with the increase in the number of keywords and many variants to consider, the advertisement management becomes a burden on even the most experienced advertisers.

The advertiser need to understand the underlying requirements of the users from the queries. These requirements are usually the descriptions of service. From the query, *create an android app for generating recipe*, the user might actually require *I want to create an android application where I can give picture of the ingredients that are available to me as an input and the generated application gives the relevant recipes out of these ingredients*. The result of the query must be relevant based on the information need present in it and not because it contains the keywords of the description.

Motivation: Advertisers have to analyze the purchase requirement of the targeted users, that helps them to aim those users with appropriate search terms. Advertises analyze query logs, search keywords reports and trend reports to determine relevant search keywords. It is very difficult to understand the latent need of the user from a few words in the query and how an advertiser characterizes an information need [1], [2]. Hence, it is necessary to develop a model on top of the search campaigns that the advertisers can examine the consequence of the topical changes in trends over the time and target new market.

Contribution: In this paper, predicting conversion in advertising using expectation maximization [PCAEM] model is proposed to provide the effectiveness of their advertising campaigns to the advertisers with respect to time period. Vocabulary is constructed from the query keywords present in search query log. Gaussians are assumed for topic assignment and likelihood function is computed to find topic distribution of a query. Topic distribution of the Gaussians with respect to time period is defined with Topic Proportion Vector. Least

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Absolute Shrinkage and Selection Operator (Lasso) is used predict total number of conversion.

Organization: This paper is organized as follows: Various probabilistic topic modeling models are studied in section 2. Expectation Maximization (EM) and Latent Dirichlet Allocation (LDA) models are discussed in section 3. Prediction Conversion in Advertising using Expectation Maximization Model and Algorithm is presented in section 4. Section 5 discussess about experiment set-up, performance metrics and results analysis. Conclusions are presented in section 6.

II. Related Works

In this section, various probabilistic models for topic modelling are reviewed. Documents topics can be discovered by clustering the similar documents. A non-parametric clustering algorithm is proposed based on local shrinking in that the number of convergent points are used as number of clusters [3]. The idea is to transform data points towards denser regions. The Shrinking process is based on K-nearest neighbour method. Value of K is selected automatically based on optimized value of Silhouette and CH index. The non-parametric clustering algorithm out-performs traditional algorithms for all given data sets even when they are provided with true number of clusters as parameters. A clustering algorithm that can identify categories from the query keywords and assign advertisement to them is proposed [4]. This algorithm is based on Bernoulli distribution. Beta priors are used to maintain discrete probability distribution to assign clusters.

Document Influence Model (DIM) model is proposed based on Dynamic Topic Model (DTM) to find topics [5] over the time. Experiments are conducted on cited documents. This model computes the sequences of topics, posterior distribution of the latent variables and the per-document influence values. It shows how past articles decide the varying influence on future articles. Article's influence value is the hidden variable and the influential articles are identified by posterior inference.

Latent Dirichlet allocation (LDA) model is widely used to assign topics to the documents and it is an unsupervised model in which words in the documents are modelled. A supervised latent Dirichlet allocation (sLDA) model [6] is introduced that accepts various response types. The response variable can be movie rating, count of number of users who selected an article important, document category. Hidden topics are discovered by combining document and response and later used to predict the response variables. The sLDA model is limited to one variable association with document. A generative labelled LDA (L-LDA) model [7] is presented with multi-label supervision. Each label is associated with one topic directly. This model is a combination of supervised LDA and mixture model of Multinomial Naive Bayes. This L-LDA model is used to assign topics to Twitter profiles correctly and find similarity of profile pairs [8]. It also re-ranks Twitter blogs and suggest new users to follow. This L-LDA performs similar to Support Vector Machine and it outperforms when training data is limited.

Search Engine Marketing [SEM] agencies manage thousands of keywords for their clients. Advertisement brokers provided a management dashboard interface that allows them to change the search campaign attributes. Advertisers then use this dashboard to create test variants for various bid choices, keywords ideas etc. Controlled experiments are performed to reveal best performing variants. As the number of keywords increases, the test variants increases and the task of campaign management becomes burdensome. The advertisers need to understand the intent of the users in order to target them with particular services. Ahmet Bulut has proposed a framework called TopicMachine [9] which enables SEMs to scale and optimize for conversions. The TopicMachine uses LDA to reveal the hidden intent of the search terms that best matches client with its users and a Lasso-based predictor that predicts the conversion.

LDA models do not find the correlations between topics. Li et al. [10] introduced Pachinko Allocation Model (PAM) which captures the relation between topics by using directed acyclic graph (DAG). Each leaf in the DAG represents a word in a vocabulary and each internal node represents a relation between either words or topics. PAM with DAG does not represent the word's topical distribution that is present in multiple topics. Hierarchical PAM (hPAM) [11] arranges topics in hierarchies. This model combines the hierarchical nature of hLDA with the topic mixing abilities of PAM. In hPAM, each node is associated with distribution over the vocabulary. The resulting model is effective at discovering mixtures of topic hierarchies.

A hierarchical algorithm which represents documents as a hierarchy of latent topics computed with Dirichlet process [12]. It is based on Bayesian priors and hierarchy of topics is derived without estimating depth of the hierarchy and branching at each level. The internal nodes represents words and topics probability distribution and vocabulary clustering is performed. Leaf nodes represents words distribution in a corpus hierarchical topic clustering. This model does not restrict on topic usage, allows multiple inheritance between topics and internal nodes are modelled as subtopics and words distribution. Method proposed in [13], [14], [15] can be used for prediction.

Ravi. et al. [16] proposed a probabilistic method that generates bid phrases for online advertising. This model first trained on search query log and generates well-formed bid phrases. Next, it generates novel bid phrases from webpages and corpus of bid phrases. Fujita et al. [17] proposed a method that generates shop-specific ad listing by incorporating promotional text data for restaurant domain. Experiments result shows that the click through rate is higher for automatically generated ad than template based ad.

III. Background

In this section, Expectation Maximization (EM) that is used to construct the proposed model and Latent Dirichlet Allocation (LDA) which is used for comparison with the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

proposed model are explained.

A. Expectation Maximization (EM)

Expectation maximization is an iterative process used to compute the maximum likelihood of parameters in a statistical model, especially in models where some data is unobserved [18], [19], [20]. It is used as a clustering technique. Each cluster can be mathematically represented as a probability distribution, characterized by its mean and variance. EM assumes that the data is generated by a mixture of underlying probability distribution. It assigns each object to a cluster depending on the likelihood of its membership. There are no restrictive boundaries between the clusters, i.e., an object can belong to multiple clusters with same or different probability of membership.

EM algorithm comprises of two distinct steps; the Expectation step ($E - Step$) and the Maximization step ($M - Step$). In the $E - step$, the estimated parameters are used to compute the likelihood of the model. In the $M - step$, the likelihood computed in the previous step are used to determine new values for the model parameters. The algorithm converges when there is no significant change in the model.

Consider two coins A and B are tossed $five$ times, the probability p of head coming up on these tosses is known as *P*=p_1, p_2...p_5. The goal is to estimate the identity of the coin for these tosses as *Q*=q_1, q_2...q_5 where q_i is the identity of the coin for the i[th] toss. Here Q is the latent or hidden variable.

For this problem, computing the proportion of heads for each coin is not a viable option. However, the parameter estimation with incomplete data can be transformed to maximum likelihood estimation with complete data using EM. The following steps are involved:

1) Initial values for P_A and P_B are selected randomly.
2) For each of the five flips, estimate the likelihood of the flip being made using coin A or B.
3) Assuming these completions to be correct, calculate the new values for parameters P_A and P_B.
4) Repeat step 2 and 3 until the difference between the previous and the current estimates is negligible. This defines the convergence point of the problem.

B. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [21] is a common method for topic modelling. It is a generative model which separates words into different topics from a *corpus*. LDA assumes that each document is a mixture of different topics, where each topic has some particular probability of generating a particular word. A document is assumed to be a *bag of words* pulled from a distribution selected by a Dirichlet process.

Consider a corpus C which is a collection of documents D. The goal is to discover K topics from C, topic distribution of D and words associated with each topic. LDA performs the following step to achieve this goal:

1) Each word W in a document D is randomly assigned to a topic T.
2) This random assignment gives a basic structure of the goal, i.e., initial topic distribution of the document and the word distribution for the topics.
3) For each word W in D, the conditional probabilities *P(T/D)* and *P(W/T)* are estimated. *P(T/D)* represents the probability of the words in D which are assigned to topic T. *P(W/T)* is the probability of W assigned to topic T over all documents in the corpus.
4) Each word is re-sampled and it is assumed that the topic assignment for the current word is incorrect. Word W is then assigned a new topic by computing *P(T/D)*p(W/T)* that gives the probability of topic T containing word W.
5) The previous step is iterated until a steady state is reached. The assignments at this state are used to estimate the topic distribution of the documents and the word distribution of the topics.

IV. Prediction Conversion in Advertising using Expectation Maximization Model and Algorithm

A. Problem Definition

Given a search log l and number of epoch e, the objective is to predict the total conversion in advertising.

B. Prediction Conversion in Advertising using Expectation Maximization Model

Predicting conversion in advertising using expectation maximization (PCAEM) model provides the advertisers with information on the effectiveness of their advertising campaigns. This effectiveness can be measured by estimating the number of conversions with respect to time period. This framework helps the advertisers in making business decisions. The PCAEM model has the following steps:

Step 1: Building a Vocabulary

In this step, the queries from the query log are considered to build a vocabulary. Stopwords *a, an, the, for etc.* are used as grammar constructs and don't convey any meaning to the query. Hence, they are removed from the queries to obtain the words that convey the requirements of the users. The vocabulary V is built from the remaining words in the queries after removal of stopwords and the frequency of occurrence of those words are computed. Consider a query *create an android app for generating recipe*, in which the words an and for are removed. After pre-processing, the query now becomes *create android app generating recipe*.

Step 2 : Gaussian Assumption for Defining Topics

A query is a mixture of various topics. Hence it is necessary to determine the latent topics in a query. Consider the query *create an android app for generating recipe*, it comprises of words related to both $technology$ as well as $food$. In EM, topics are represented as Gaussians *Gauss[K]* , where K is the number of topics. A Gaussian is characterized by

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

its mean and variance. Gaussians contain all the words in the vocabulary, in which each word is randomly assigned a topic between 1 to K. The initial probabilities of the words belonging to topic i in Gaussians are equal, where $i \in 1, ..k$. The initial mean of a Gaussian and covariance between the Gaussians is computed as shown in Function 1.

Function 1: Initial Mean and Covariance Generation

Function: InitialTopic

Data: Consider Vocabulary V and Number of Topics K. Initial Mean and Covariance Matrix are generated.

Let n = number of Gaussians = number of topics

$mean[n][K]$ = mean of each topic in K for n Gaussian

$covariance[n][n]$ = covariance between Gaussians

$Gauss[n]$ = n Gaussians

V_{size} = vocabulary size

for *i = 1 to n* **do**
- $Gauss[i]$ = All vocabulary words
- **for** *j = 1 to* V_{size} **do**
  - $Gauss[i].word[j]$ = random topic assignment between 1 to K
- **for** *L = 1 to K* **do**
  - Let Num_{word} = Number of words in Gaussian i with topic K
  - Let Tot_{word} = Total number of words in Gaussian i
  - Calculate mean for topic L in $Gauss[i]$ = $Mean[i][L] = \frac{Num_{word}}{Tot_{word}}$

for *i = 1 to n* **do**
- **for** *j = 1 to n* **do**
  - Calculate covariance(Gauss[i],Gauss[j]) using Equation 1

Here, $Prob_{Gauss_i}$ is the probability of $word_c$ in Gauss[i], $Prob_{Gauss_j}$ is the probability of $word_c$ in Gauss[j], $topic(word_c,i)$ is the topic $word_c$ in Gaussian i and $mean[i][topic(word_c,i]$ is the mean of $topic(word_c,i)$ in Gaussian i.

Step 3 : Compute Likelihood to find Topic Distribution of a Query

In the previous step, the topics defined in the Gaussians are estimated. Using these Gaussians, the topic distribution of the queries is computed. All the queries are assumed to be stationary points in a space. This space also contains the Gaussians with equal probability $P(c)$. As the Gaussians move around and take shape, the conditional probability of occurrence $P(qi/c)$ of a query qi in the Gaussian c changes. A query qi can belong to more than one topic. Consider the query *create an android app for generating recipe*, the words $android$ and app come from the $technology$ topic and the word $recipe$ comes from the $food$ topic, hence this query occurs in the $technology$ topic with probability $(2/5)$ and in the $food$ topic with probability $(1/5)$. The likelihood of a query occurring in a topic is computed as shown in Function 2.

$$covariance_c(j,k) = \sum_{j=1}^{Q_{num}} \left(\frac{PCQ(c,i)}{nP(c)}\right) * \Big(q_{i,j} - mean(c,j)\Big) * \Big(q_{j,k} - mean(c,k)\Big) \quad (6)$$

Function 2: Query Topic Likelihood

Function: QueryTopicLikelihood

Data: Consider Mean matrix $mean[n][k]$, Covariance matrix $cov[n][n]$, Number of queries Q_{num} and n is the number of Gaussians. Conditional Probability of Topic C for given Query q PCQ, Conditional Probability of Query q for given Topic C PQC and Probability of Gaussian $P(C)$ are generated.

for *i = 1 to* Q_{num} **do**
- **for** *c = 1 to K* **do**
  - Compute PQC(*i*,*c*) using Equation 2

for *c = 1 to K* **do**
- **for** *i = 1 to* Q_{num} **do**
  - Compute PCQ(*c*,*i*) using Equation 3

for *c = 1 to n* **do**
- Compute P(*c*) using Equation 4

Here, $\mid cov_c \mid$ is the determinant of $covariance_c$.

$$PCQ(c,i) = \frac{PQC(i,c) * P(c)}{\sum_{c'=1}^{n} PQC(i,c') * P(c')} \quad (3)$$

$$P(c) = \frac{1}{n} \sum_{j=1}^{Q_{num}} PCQ(c,j) \quad (4)$$

Step 4 : Compute New Definition of Topics

The initial mean and covariance of the Gaussians give a basic structure to the topic. The conditional probabilities estimated in the previous step influence the mean and covariance of the Gaussian such that the mean of the topics in the Gaussian changes. Hence, the Gaussians need to be recomputed with respect to the conditional probabilities. The mean and covariance of the Gaussians is computed using Equation 5 and 6.

$$mean(c,j) = \sum_{i=1}^{Q_{num}} \left[\frac{PCQ(c,i)}{cP(c)}\right] q_{i,j} \quad (5)$$

Step 5 : Test for Convergence of the Model

Convergence signifies a stable state of the model. After certain number of iterations the Gaussians gain a definite shape and position in space where the change in the conditional

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

$$covariance_c(i,j) = [(Prob_{Gauss_i}) - (mean[i][topic(word_c, i))] * [(Prob_{Gauss_j}) - (mean[j][topic(word_c, j))] \quad (1)$$

$$PQC(i,c) = \frac{1}{\sqrt{2\pi \mid cov_c \mid}} \left\{ \frac{-1}{2} \left[\sum_{K=1}^{n} \sum_{j=1}^{n} ((q_{i,j} - mean(c,j)) * (q_{i,k} - mean(c,k))) * \left(cov_c\right)^{-1}_{j,k} \right] \right\} \quad (2)$$

probabilities is negligible. Hence, step 3 and 4 need to be iterated until the model converges.

Step 6 : Compute Topic Proportion Vector
Topic proportion vector TPV represents the topic distribution of the Gaussians in the model. It is obtained from the mean computed at the convergence. As this model predicts the total number of conversions for advertisement campaign in a particular time period which is called an $epoch$. TPV for each epoch is defined as $TPV[e][i]$, where e and i represent epoch and topic respectively. TPV for an epoch represents the weight of the topics in that epoch.

Step 7 : Conversion Prediction
A conversion occurs when a user clicks on the advertisement and performs some action that gives benefit to the advertiser. This type of user click is defined as Cost Per Click (CPC). In order to predict the total number of conversions, the model is trained using the L_1 *prior* also known as $Lasso$ (Least Absolute Shrinkage and Selection Operator) [22]. The conversion for each epoch is estimated using Equation 7.

$$conversion = \frac{1}{2 * T} * \mid\mid CPC - TPV * R \mid\mid_2^2 + \alpha * \mid\mid R \mid\mid_1 \quad (7)$$

Here, α is the hyper parameter, R is the Regression and CPC is the cost per click.

Algorithm 1: Predicting Conversion in Advertising using Expectation Maximization

Input : Query log l, Number of epoch N_e
Output: Conversion prediction cp

begin
- Pre-processing :
- Build vocabulary V by considering queries from log l
- **for** *m = 1 to* N_e **do**
  - Assume K Gaussians to represent topics as explained in step 2.
  - **while** *model not converged* **do**
    - Estimate the likelihood to find topic distribution of each query as per step 3.
    - Re-estimate Gaussians to find new definition of the topics as per step 4.
  - **for** *n = 1 to K* **do**
    - Construct Topic Proportion Vector $TPV[m][n]$ using the final Gaussian estimation after convergence.
- Prediction :
- Predict conversions using $LASSO$ method as per step 7

C. Algorithm

Predicting conversion in advertising using expectation maximization (PCAEM) algorithm as shown in Algorithm 1 has two phases : Pre-processing and Prediction. Pre-processing involves building vocabulary, Gaussian assumption, computing likelihood for topic distribution of queries, re-estimation of the Gaussians and constructing topic proposition matrix. Prediction involves Lasso based conversion prediction.

V. Experiments

A. Data Collection

In this experiment, the same dataset as [9] is used. The data is collected from U.S ad U.K market from over a 34 week period.It consists of various queries ranging over multiple topics. This dataset consists of following information: Max. CPC, Keyword,Average position, Average CPC, Clicks, CTR, Cost and Impressions. Here, Click Through Rate (CTR) is the ratio of how often user clicks an advertisement that appears as a relevant result for the query. It evaluates the competence of the keywords and the advertisement. An impression signifies the relevance of an advertisement and is incremented every time it appears as a result for a searched query. There were four different campaigns with 13,898 unique search terms that resulted in 432 conversions in total; 29,821 clicks were received out of a total of 2,382,317 ad impressions. In the proposed model, keywords are used to generate the Topic Proportion Matrix and CPC influence the conversion. Hence, Keyword and Average CPC is used in this experiment.

B. Experiment Setup

The proposed model PCAEM uses probabilistic estimations for topic modelling. LDA also works with same principle, hence PCAEM is compared with TopicMachine [9].

The setup of PCAEM is as follows: Vocabulary is constructed from the keywords. The words in the vocabulary are ranked based on their frequency of occurrence. Consider the query *create android application for food suggestions*; this query gives outcomes related to existing applications that make food suggestions rather than tutorials to learn android programming. In this example, the word *create* is a

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

low frequency word. Thus, it can be concluded that words with low frequencies do not contribute significantly the topic proportion of a query thereby influencing the outcome. Hence, the size of the vocabulary is restricted to 500. Experiments are conducted with varying topic number *K = 5, 6, 7, 8, 9, 10, 11*. The initial probability of each topic is considered equal. Using these assumptions the topic distribution of each query is estimated. The model is iterated 10 times for convergence. It is observed that PCAEM model converges at the fifth iteration. Hence, the number of conversions are estimated after the fifth iteration. The data is collected for 34 weeks, hence the number of epochs are 34 as 1 epoch is for 1 week. The experiment is conducted by varying epoch size to 15, 20, 25, 30 and 34. Topic proportion vector is constructed for each epoch and used for conversion prediction.

The setup for TopicMachine is as follows: Vocabulary is constructed in a similar way as PCAEM. Initially, words in the queries are randomly assigned to topics. These assignments are used to approximate hidden variables using variational approximation iteratively. Variation threshold is set to define a convergence point for the model. Experiments are conducted by varying the threshold parameter to 0.1, 0.01 and 0.001. It is observed that definite topics are obtained when threshold is set to 0.001. Topic proportion vector is constructed for each epoch and used for conversion prediction.

C. Performance Metrics

The performance metrics used for comparison are Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Predictive R^2 i.e., the coefficient of determination. The actual value for each epoch corresponds to the average CPC collected from the dataset. Consider E is the total number of epochs, dataset is partitioned into training set T and testing set Te, then $offset$ = E * $Te/100$. The predicted and actual value is denoted by p and a respectively.

RMSE is the root of the square of the difference between the values predicted by the model and the actual values as shown in Equation 8. MAE is the average of the absolute value of the difference between the values predicted by the model and the actual values as shown in Equation 9. Coefficient of determination is computed as shown in Equation 10. It measures the correctness of the model based on the proportion of deviation of predicted value from actual value.

$$RMSE = \sqrt{\sum_{i=E-offset+1}^{E} (a_i - p_i)^2} \tag{8}$$

$$MAE = \frac{1}{N} \sum_{i=E-offset+1}^{E} \mid p_i - a_i \mid \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=E-offset+1}^{E} (a_i - p_i)^2}{\sum_{i=E-offset+1}^{E} (a_i - mean(p_i))^2} \tag{10}$$

D. Performance Evaluation

In this section, experiment results are presented and discussed. Performance metrics are used to compare the results of the proposed model PCAEM and TopicMachine [9]. Experiments are conducted by varying training dataset to 70%, 75%, 80%, 85%, 90% and testing dataset to 30%, 25%, 20%, 15%, 10%. Experiments have been conducted on 4GB memory and Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz processor. The reproduction of TopicMachine does not get the same result as in [9], this discrepancy is mainly due to the system properties of the machine used and the programming language; all development is done in Java. Dataset used in the experiments for PCAEM and TopicMachine are the same as discussed in data collection. The performance is evaluated by computing the average of 20 independent runs.

RMSE, MAE and R^2 by Varying Number of Topic K

In order to measure the effect of the number of topics K, all the performance metrics are computed to measure the performance of PCAEM and TopicMachine. The value of K varies from 5 to 11. Table I shows the comparison of RMSE, MAE and R^2 values for both the methods by setting vocabulary size to 500, α to 0.01, epoch to 34 and training dataset to 85%. It is observed from the table that the number of topics does not affect much on the performance of PCAEM and TopicMachine. It is also observed from the table that the RMSE and MAE values are lesser in PCAEM than TopicMachine, hence the performance of PCAEM is better than TopicMachine.

RMSE, MAE and R^2 by Varying Training Dataset

In order to analyze the the performance of models with available training dataset, RMSE, MAE and R^2 is computed by increasing training dataset. Table II shows the comparison of RMSE, MAE and R^2 values for both the methods by setting vocabulary size to 500, α to 0.01, Topic number K to 10 and epoch to 34. It is observed from the Table II that as the training dataset increases the RMSE decreases and MAE increases marginally.

RMSE, MAE and R^2 by Varying epoch Size

In order to study the consequence of the size of the dataset on the performance of PCAEM and TopicMachine, RMSE, MAE and R^2 is computed by varying the epoch size. Table III shows the comparison of RMSE, MAE and R^2 values for both the methods by setting vocabulary size to 500, α to 0.01, Topic number K to 10 and training dataset to 85%. There is not much difference observed by varying size of the dataset.

PCAEM and TopicMachine Sensitivity to Hyper Parameter α

The hyper parameter α is used for prediction conversion, it is necessary to study the effect of changes in α affect the quality of model. RMSE, MAE and R^2 is computed by varying the α value for both the models. It is observed from Table II that RMSE values decreases when training dataset increases, hence, experiments are conducted by setting training dataset 90% and 85%. Table IV shows the comparison of RMSE, MAE and R^2 values for both the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE I: RMSE, MAE and R^2 by Varying Number of Topic K for PCAEM and TopicMachine. The vocabulary size, α, epoch and training dataset are set to 500, 0.01, 34 and 85% respectively

	PCAEM Method			TopicMachine		
Topic	RMSE	MAE	R^2	RMSE	MAE	R^2
5	8.6142	4.3039	-18.6755	8.6898	4.3416	-19.0221
6	8.6071	4.3004	-18.6431	8.6820	4.3377	-18.9863
7	8.6054	4.2995	-18.6354	8.6839	4.3387	-18.9949
8	8.6040	4.2988	-18.6286	8.6838	4.3386	-18.9948
9	8.6034	4.2985	-18.626	8.6794	4.3364	-18.9743
10	8.6019	4.2978	-18.6193	8.6795	4.3365	-18.9751
11	8.6015	4.2975	-18.6172	8.6706	4.3321	-18.934

TABLE II: RMSE, MAE and R^2 by Varying Training dataset for PCAEM and TopicMachine. The vocabulary size, α, Topic Number K and epoch are set to 500, 0.01, 10 and 34 respectively

	PCAEM Method			TopicMachine		
Training Dataset	RMSE	MAE	R^2	RMSE	MAE	R^2
70	12.2351	4.0718	-6.1212	12.3420	4.1073	-6.2462
75	10.9441	4.1300	-7.7055	11.0403	4.1662	-7.8592
80	9.4557	4.2235	-11.8799	9.5403	4.2612	-12.1114
85	8.6019	4.2978	-18.6193	8.6795	4.3365	-18.9751
90	6.2202	4.3970	-40.7745	6.2770	4.4370	-41.5397

TABLE III: RMSE, MAE and R^2 by Varying epoch Size for PCAEM and TopicMachine. The vocabulary size, α, Topic Number K and training dataset are set to 500, 0.01, 10 and 85% respectively

	PCAEM Method			TopicMachine		
epoch	RMSE	MAE	R^2	RMSE	MAE	R^2
15	3.4425	3.4425	0.8266	3.5394	3.5394	0.8168
20	6.7551	4.7729	-19.9357	6.8263	4.8230	-20.379
25	5.0849	3.5898	-0.4554	5.1259	3.6187	-0.4790
30	6.7661	3.9063	-3.0550	6.6234	3.9394	-3.1240
34	8.6019	4.2978	-18.6193	8.6795	4.3365	-18.9751

methods by setting vocabulary size to 500, Topic number K to 10, epoch to 34 and training dataset to 90%. Table V shows the comparison of RMSE, MAE and R^2 values for both the methods by setting vocabulary size to 500, Topic number K to 10, epoch to 34 and training dataset to 85%. It is observed from the Table IV and V that PCAEM model is sensitive to hyper parameter α, but TopicMachine is not. PCAEM model is giving best result when α is set to 10.

It is observed from Table IV and V that PCAEM model is giving best results when α is set to 10. Hence, RMSE and MAE is computed again by varying training dataset, by varying the number of the topic K and by varying the $epoch$ size by setting α value to 10. Figures 1 and 2 show the comparison of RMSE and MAE by Varying Training Dataset when α is set to 10 respectively. It is observed that the average value of RMSE is 4.4572 and 9.5752 of PCAEM and TopicMachine respectively. The average value of MAE is 1.9346 and 4.2614 of PCAEM and TopicMachine respectively.

Figures 3 and 4 show the comparison of RMSE and MAE by the varying number of topics K when α is set

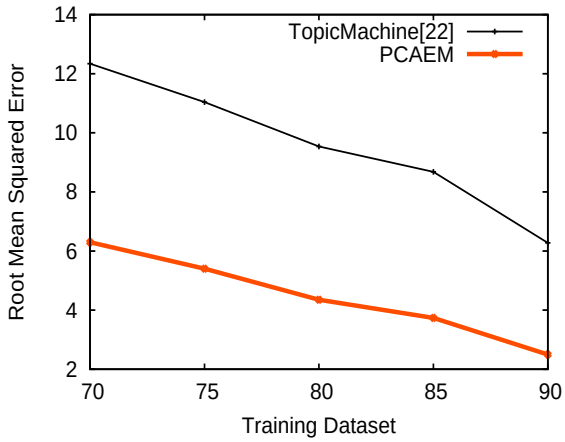

Fig. 1: RMSE by Varying Training Dataset When α = 10

to 10 respectively. It is observed that the average value of RMSE is 3.2013 and 8.6815 of PCAEM and TopicMachine respectively. The average value of MAE is 1.5861 and 4.3375 of PCAEM and TopicMachine respectively.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE IV: RMSE, MAE and R^2 by Varying Hyper Parameter α for PCAEM and TopicMachine. The vocabulary size, Topic Number K, epoch and training dataset are set to 500, 10, 34 and 90% respectively

	PCAEM Method			TopicMachine		
α	RMSE	MAE	R^2	RMSE	MAE	R^2
100	80.9580	57.2459	-7075.36	6.2769	4.4370	-41.5388
50	37.3657	26.4213	-1506.43	6.2778	4.4377	-41.5518
30	19.9290	14.0915	-427.807	6.2771	4.4371	-41.541
20	11.2110	7.9266	-134.7	6.2771	4.4372	-41.542
10	**2.4962**	**1.7617**	**-5.7279**	6.2773	4.4373	-41.5442
1	5.3574	3.7867	-29.9892	6.2772	4.4372	-41.5429
0.1	6.1418	4.3415	-39.7275	6.2778	4.4376	-41.5508
0.01	6.2202	4.3970	-40.7745	6.2778	4.4376	-41.551
0.001	6.2281	4.4025	-40.8799	6.2774	4.4374	-41.5461

TABLE V: RMSE, MAE and R^2 by Varying Hyper Parameter α for PCAEM and TopicMachine. The vocabulary size, Topic Number K, epoch and training dataset are set to 500, 10, 34 and 85% respectively

	PCAEM Method			TopicMachine		
α	RMSE	MAE	R^2	RMSE	MAE	R^2
100	114.6907	57.3451	-3486.75	8.6793	4.3364	-18.974
50	53.0421	26.5205	-744.988	8.6791	4.3362	-18.9728
30	28.3834	14.1907	-212.609	8.6790	4.3362	-18.9723
20	16.0551	8.0258	-67.3464	8.6797	4.3366	-18.976
10	**3.7365**	**0.8609**	**-2.7018**	8.6794	4.3364	-18.9746
1	7.3823	3.6874	-13.4503	8.6794	4.3365	-18.9746
0.1	8.4910	4.2423	-18.1167	8.6797	4.3366	-18.9758
0.01	8.6019	4.2978	-18.6193	8.6793	4.3364	-18.9738
0.001	8.6130	4.3033	-18.6699	8.6796	4.3365	-18.9755

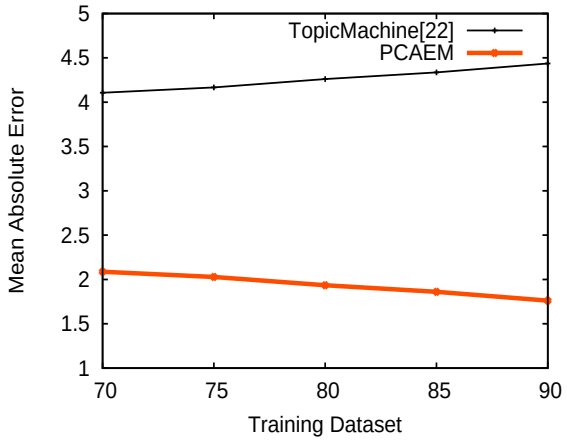

Fig. 2: MAE by Varying Training Dataset When α = 10

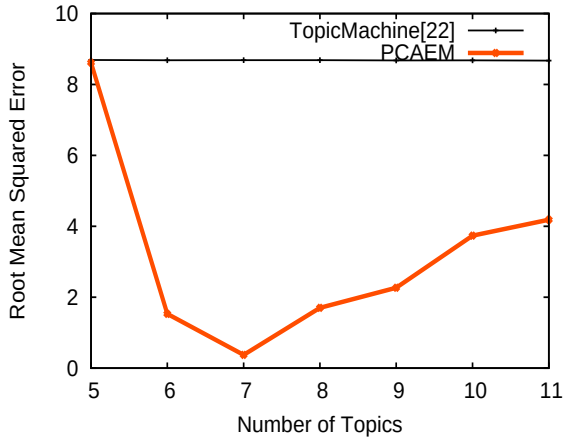

Fig. 3: RMSE by Varying Number of Topics When α = 10

Figures 5 and 6 show the comparison of RMSE and MAE by varying the $epoch$ size when α is set to 10 respectively. It is observed from that the average value of RMSE 3.1952 and 6.1987 of PCAEM and TopicMachine respectively. The average value of MAE is 2.1568 and 4.0513 of PCAEM and TopicMachine respectively.

VI. CONCLUSIONS

In this work, we present predicting conversion in advertising using expectation maximization [PCAEM] model to understand advertising campaigns' effectiveness to the advertises over the time. Search query log is used to build vocabulary. Expectation Maximization method is used to find the hidden topics and topic distribution on search terms. Least Absolute Shrinkage and Selection Operator (LASSO) is used predict total number of conversion. Experiments are performed on

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

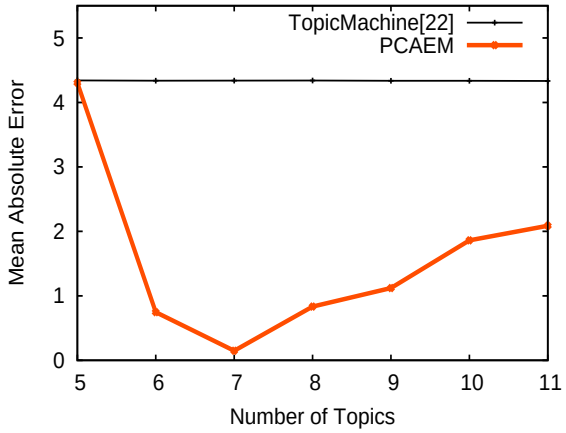

Fig. 4: MAE by Varying Number of Topics When α = 10

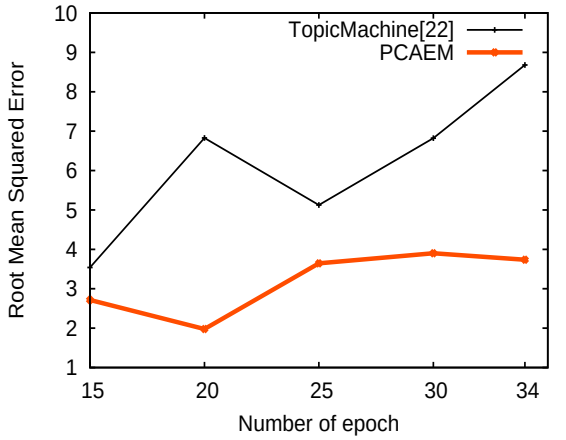

Fig. 5: RMSE by Varying *epoch* When α = 10

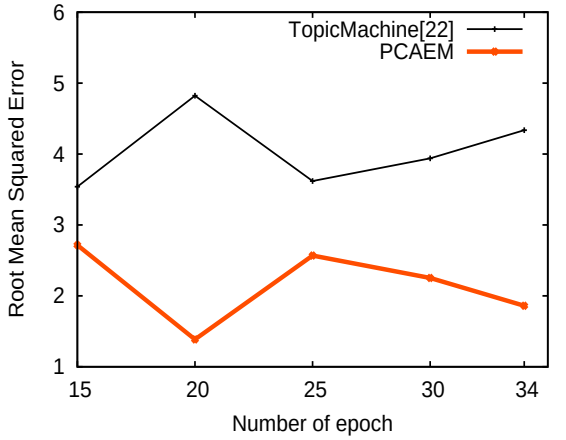

Fig. 6: MAE by Varying *epoch* When α = 10

query data used in [9] which is collected from the U.K and the U.S market over 34 weeks. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Predictive R^2 are used as performance metrics. Experiment results are compared with TopicMachine [9]. The proposed method outperforms TopicMachine [9] by reducing RMSE and MAE. The PCAEM model is sensitive to hyper parameter used for prediction conversion while TopicMachine is not.

REFERENCES

[1] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. Silva de Moura, "Impedance Coupling in Content-targeted Advertising," *In the Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 496–503, 2005.

[2] P. Rusmevichientong and D. P. Williamson, "An Adaptive Algorithm for Selecting Profitable Keywords for Search-based Advertising Services," *Proceedings of the 7th ACM Conference on Electronic Commerce*, pp. 260–269, 2006.

[3] X. Wang, W. Qiu, and R. H. Zamar, "An Iterative Non-parametric Clustering Algorithm based on Local Shrinking," *Computational Statistics and Data Analysis*, vol. 52, pp. 286–298, 2007.

[4] A. Schwaighofer, J. Q. Candela, T. Borchert, T. Graepel, and R. Herbrich, "Scalable Clustering and Keyword Suggestion for Online Advertisements," *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pp. 27–36, 2009.

[5] S. Gerrish and D. M. Blei, "A Language-based Approach to Measuring Scholarly Impact," *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 375–382, 2010.

[6] J. D. Mcauliffe and D. M. Blei, "Supervised Topic Models," *Advances in Neural Information Processing Systems*, pp. 121–128, 2008.

[7] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora," pp. 248–256, 2009.

[8] D. Quercia, H. Askham, and J. Crowcroft, "TweetLDA: Supervised Topic Classification and Link Prediction in Twitter," in *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 2012, pp. 247–250.

[9] A. Bulut, "TopicMachine: Conversion Prediction in Search Advertising using Latent Topic Models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2846–2858, 2014.

[10] W. Li and A. McCallum, "Pachinko Allocation: DAG-structured Mixture Models of Topic Correlations," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 577–584, 2006.

[11] D. Mimno, W. Li, and A. McCallum, "Mixtures of Hierarchical Topics with Pachinko Allocation," *Proceedings of the 24th International Conference on Machine Learning*, pp. 633–640, 2007.

[12] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Non-parametric Estimation of Topic Hierarchies from Texts with Hierarchical Dirichlet Processes," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2749–2775, 2011.

[13] K. Srinivasa, K. Venugopal, and L. M. Patnaik, "An Efficient Fuzzy based Neuro–genetic Algorithm for Stock Market Prediction," *International Journal of Hybrid Intelligent Systems*, vol. 3, no. 2, pp. 63–81, 2006.

[14] K. Srikantaiah, M. Roopa, N. K. Kumar, K. Venugopal, and L. Patnaik, "Automatic discovery and ranking of synonyms for search keywords in the web," *International Journal of Web Science*, vol. 2, no. 4, pp. 218–236, 2014.

[15] K. Srikantaiah, N. K. Kumar, K. Venugopal, and L. M. Patnaik, "Web Caching and Prefetching with Cyclic Model Analysis of Web Object Sequences," *International Journal of Knowledge and Web Intelligence 2*, vol. 5, no. 1, pp. 76–103, 2014.

[16] S. Ravi, A. Broder, E. Gabrilovich, V. Josifovski, S. Pandey, and B. Pang, "Automatic Generation of Bid Phrases for Online Advertising," *Proceedings of the Third ACM International Conference on Web Search and Data mining*, pp. 341–350, 2010.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[17] A. Fujita, K. Ikushima, S. Sato, R. Kamite, K. Ishiyama, and O. Tamachi, "Automatic Generation of Listing Ads by Reusing Promotional Texts," *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business*, pp. 179–188, 2010.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[19] J. A. Bilmes *et al.*, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *Journal on International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[20] T. K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[22] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500