Rank Prediction in graphs with Locally Weighted Polynomial Regression and EM of Polynomial Mixture Models

Michalis Rallis
Athens University of Economics and Business
A.U.E.B
Greece
r.mixalis@gmail.com

Michalis Vazirgiannis
Athens University of Economics and Business
A.U.E.B
Greece
mvazirg@aueb.gr

***Abstract*— In this paper we describe a learning framework enabling ranking predictions for graph nodes based solely on individual local historical data. The two learning algorithms capitalize on the multi feature vectors of nodes in graphs that evolve in time. In the first case we use weighted polynomial regression (LWPR) while in the second we consider the Expectation Maximization (EM) algorithm to fit a mixture of polynomial regression models. The first method uses separate weighted polynomial regression models for each web page, while the second algorithm capitalizes on group behavior, thus taking advantage of the possible interdependence between web pages. The prediction quality is quantified as the similarity between the predicted and the actual rankings and compared to alternative baseline predictor. We performed extensive experiments on a real world data set (the Wikipedia graph). The results are very encouraging.**

Keywords; Expectation-Maximization, Clustering, Mixture Models, Maximum Likelihood Estimation, Locally Weighted Regression, Polynomial Regression

I. INTRODUCTION

Large and evolving graphs constitute an important element in current large scale information systems. Common cases of such graphs are the Web graph, social networks, citation graphs, CDRs (call data records) where nodes (featured with attributes) are connected to each other with directed edges representing endorsement/recommendation/friendship. The most prominent such example is the ranking of web/pages nodes in the Web Graph where the ranking – based on queries most of the times – is of huge importance as there is a whole industry working on optimizing the content and linking structures of web pages to achieve top rankings in specific queries. The ranking of nodes in graphs usually represents its centrality and authority with regards to the linking structure and also capitalizes on a wealth of node features related to their content and to the graph linking patterns. Another inherent and interesting feature of such graphs is their dynamism and fast evolution as new nodes join the graphs while the linking structure is very volatile.

On the other hand the owner of the individual node (be it a blog, a web page, an author in a citation graph, a participant in a social network) can see her ranking only in the case of the web graph by submitting queries to the owner of the graph (i.e. to a search engine). Assuming a series of time-ordered rankings of the nodes of a graph where each node bears a vector with the values of its features for each time stamp, we develop learning mechanisms that enable predictions of the nodes' ranking in future times. We stress that the predictions require only local feature knowledge while no global data need to be known. In such a case the node could a. plan actions for optimizing its ranking in the future and b. organize advertising campaigns, etc.

The first algorithm is trained from a time sequence of preprocessed rank values using a separate weighted polynomial regression model. So, it only takes advantage of the pattern that a specific page follows. On the other hand, in the second approach we group pages into clusters and fit a mixture of polynomial regression models in an effort to take advantage of the correlation of the evolution pattern of different web-pages. Both of the algorithms perform the learning procedure considering that each web-page follows a time-sequence of ranking values (target variable) and each ranking value is explained by a multi feature vector. Our contributions lie in the following: **i**. a learning framework enabling ranking predictions for graph nodes based solely on individual local historical data, **ii**. an application of the EM principles on a mixture of polynomial regression models for graph based structures, **iii**. a thorough experimental evaluation on a large scale real time evolving graph (Wikipedia).

II. RELATED WORK

The methods used in this paper have been proposed in several other works such as clustering time sequences. Specifically, Gaffney/Smyth [8] used the EM algorithm to fit a mixture of regression models to cluster data that evolve in time such as video streams. Lall et al. [10] proposed a locally weighted polynomial regression model as well as methods for choosing the model parameters and conducted experiments in

time series data of the Great Salt Lake. Also, O'Madadhain et al. [11] proposed methods of time series analysis in event based graph structures such as e-mail, telephone calls and research publications in an effort to address the problems of predicting future co-participation of entities and the future rank of authority, influence, etc.

To prove the effectiveness of their models they used classification algorithms such as Logistic Regression and Naive Bayes. However, in our work we evaluate the effectiveness of LWPR and mixture of polynomial regression models in predicting future rank values for graph based structures.

Related work regarding page rank predictions are presented in [12] [13] [14]. In [12] and [14] authors use Markov Models with page ranking and an alternative of the Google's PageRank algorithm, to produce personalized predictions considering the path of a specific user. However in these efforts, unlikely to this work and that reported in [11], data are not considered as time series, instead they are represented as a static network or graph for which a rank is produced.

Finally, in [1] authors developed a linear regression model and EM to fit a mixture of linear regression models to predict the future page rankings, yet they have not considered higher order polynomials and multi feature vectors in their proposed models.

III. METHODS

A. Locally Weighted Polynomial Regression (LWPR)

In order to predict future rankings of Web pages, we need to define a measure that effectively expresses the trends of Web pages among different snapshots of the Web graph. We have adopted a measure (*racer*) we introduced in [6] suitable for measuring page rank dynamics.

Consider the case of a web-page whose successive rankings in time produce a time-sequence of *racer* values: $\{x_1, x_2, x_3, ..., x_m\}$, m the number snapshots used for the algorithm training. Also, for each timestamp we maintain a multi feature vector $f^{(c)} =< f_1^{(c)}, f_2^{(c)}, ..., f_p^{(c)} >$ where c represents a specific snapshot and p is the number of different attributes.

Let $\vec{\theta} =< \theta_1, \theta_2, ..., \theta_p >$ represent the parameters of the regression model, then our goal is to fit **θ** in order to minimize the error function "(1)" of a specific page.

$$J(\theta) = \frac{1}{2}\sum_{c=1}^{m} w^{(c)} \left(x_c - \theta^{\mathrm{T}} f^{(c)}\right)^2 \quad (1)$$

The weight terms should be large when our new point, for which we have to predict a racer value, is close to the training example $f^{(c)}$ and small when the opposite holds. So a reasonable choice for the weights is:

$$w^{(c)} = \exp\left(-\frac{(f^{(c)} - f)^T (f^{(c)} - f)}{2r^2}\right) \quad (2)$$

where f is the new feature vector used to predict the future racer value and r is the bandwidth parameter which controls how fast the weight falls of with the distance between $f^{(c)}$ and f. Assuming that for a vector v, we have $v^T v = \sum_i v_i^2$ then the objective function can be written in matrix notation as:

$$J(\theta) = \frac{1}{2}(F\theta - \vec{x})^T W (F\theta - \vec{x}) \quad (3)$$

$$\text{where W} = \text{diag}\,(w^{(1)}, w^{(2)}, .., w^{(m)})$$

Each $w^{(c)}$ $c = 1..m$ is defined as in "(2)". Matrix F contains in each row the corresponding multi feature vector $f^{(c)}$, with the first element being 1 to represent the intercept term. Then, **θ** is a column vector containing the model parameters, and **x** is a column vector containing the corresponding racer values for each training example.

In order to minimize the objective we take the gradient of *J* with respect to **θ** and set it to zero:

$$\nabla_\theta ((F\theta - \vec{x})^T W (F\theta - \vec{x})) = 0 \quad (4)$$

The value of **θ** that minimizes *J(θ)* is:

$$\theta = (F^T W F)^{-1} F^T W \vec{x} \quad (5)$$

In order to obtain a better fit to the training data we used a polynomial regression model, which has the form of:

$$x_c = \theta_0 f_1^{(c)} + \theta_1 f_2^{(c)} + \theta_2 f_3^{(c)} + .. + \theta_l f_l^{(c)} + ... +$$

where we consider all possible combinations of feature attributes with repetition. Specifically, if we assume a model with *d*-dimensional feature vectors and polynomial degree *k* then we should also add:

$C^*(d,k) = \frac{(d+k-1)!}{k!(d-1)!}$ attributes with their corresponding parameters.

B. Mixture of Polynomial Regression Models

We assume that each web-page belongs to one of *J* different clusters with $J << n$, with *n* being the number of all pages. Specifically, we consider *J* polynomial regression models, each governed by each own parameter vector θ_j $j \in \{1, ..., J\}$. We also consider a different variance for each component: σ_j^2. To generate the racer values $x^{(i)}$ of the i-th web-page we first choose cluster j with probability π_j where $\sum_{j=1}^{J} \pi_j = 1$ and then estimate the values $x^{(i)}$ as:

$$x_c^{(i)} = h_{\theta_j}(f_i^{(c)}),\ c = 1...m$$

where the hypothesis function $h_{\theta_j}(f_i^{(c)})$ represents a polynomial regression model parameterized by the weights of component j with m being the number of available snapshots for training the model. So, given a cluster j the racer values of the i-th page are drawn from the following product of Gaussians:

$$p(x^{(i)} \mid j) = \prod_{c=1}^{m} N(x_c^{(i)} \mid h_{\theta_j}(f_i^{(c)}), \sigma_j^{\,2}) \qquad (6)$$

We can compute the model parameters $\sigma_\kappa, \pi_\kappa, \theta_0^{\,\kappa}, \theta_1^{\,\kappa}, ..., \theta_p^{\,\kappa}$, $\kappa = 1..J$, by maximizing the data log likelihood $L(\theta)$ using the EM approach. Specifically we need to *maximize*:

$$L(\theta) = \sum_{i=1}^{n} \log p(x^{(i)}) =$$

$$\sum_{i=1}^{n} \log \sum_{j=1}^{J} \pi_j \, p(x^{(i)} \mid j) =$$

$$\sum_{i=1}^{n} \log \sum_{j} \pi_j \left(\frac{1}{2\pi}\right)^{m/2} \sigma_j^{\,-m} \exp\left\{ \frac{-\sum_{c=1}^{m} (x_c^{(i)} - h_{\theta_j}(f_i^{(c)}))^2}{2\sigma_j^2} \right\}$$

subject to:

$$\sum_{j=1}^{J} \pi_j = 1$$

By taking the derivatives of the above equation and setting them to zero we obtain the maximum likelihood estimates, which we use to execute the *maximization step*:

$$\sigma_\kappa^2 = \frac{\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} \left(x_c^{(i)} - h_{\theta_\kappa}\left(f_i^{(c)}\right)\right)^2}{m \sum_{i=1}^{n} p(\kappa \mid x^{(i)})} \qquad (7)$$

$$\pi_\kappa = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_\kappa \, p(x^{(i)} \mid k)}{\sum_{j} \pi_j \, p(x^{(i)} \mid j)} \qquad (8)$$

We can find the parameters $\theta_0^{\kappa}, \theta_1^{\kappa}, ..., \theta_p^{\kappa}$ by taking the derivatives of the log likelihood function for each parameter and solving a system of equations which can be written in matrix notation in the following form:

$$\theta_\kappa = \mathrm{X}^{-1}\mathrm{B} \qquad (9)$$

where **X** is the following matrix:

$$\begin{matrix}
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} 1 & ..... & \sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} f_{pi}^{(c)} \\
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} f_{1i}^{(c)} & ..... & \sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} f_{pi}^{(c)} f_{1i}^{(c)} \\
..... & ..... & ..... \\
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} f_{pi}^{(c)} & ..... & \sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} f_{pi}^{(c)} f_{pi}^{(c)}
\end{matrix}$$

and B:

$$\begin{matrix}
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} x_c^{(i)} \\
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} x_c^{(i)} f_{1i}^{(c)} \\
..... \\
.... \\
\sum_{i=1}^{n} p(\kappa \mid x^{(i)}) \sum_{c=1}^{m} x_c^{(i)} f_{pi}^{(c)}
\end{matrix}$$

Instead of using random values, we employed K-Means [7] clustering algorithm to initialize the parameters of EM. To specify the number of clusters to use, we executed the algorithm iteratively for different number of clusters and different times for each cluster size in order to obtain a significant reduction in the distortion measure:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} \{x^{(i)} \in C_k\} \parallel x^{(i)} - \mu_\kappa \parallel^2$$

where N the number of data points to cluster (in our problem this is the number of racer value sequences each one corresponding to a web-page), K the number of clusters, C_κ the set of points grouped in cluster κ, $x^{(i)}$ the sequence of racer values for the i-th page and μ_κ the centroid of cluster κ.

Finally, during the expectation step we calculate the conditional probabilities:

$$p(\kappa \mid x^{(i)}) = \frac{\pi_\kappa \, p(x^{(i)} \mid \kappa)}{\sum_{j} \pi_j \, p(x^{(i)} \mid j)} \quad \forall \kappa = 1..J, \forall i = 1..N \qquad (10)$$

After having estimated the model parameters, in order to make future predictions for a given page i we use the theta (**θ**) values of the component that receives the highest probability given the page i. Particularly, the racer value for the i-th page at time t is: $x_t^{(i)} = h_{\theta_\kappa}(f_i^{(t)})$ and $\kappa = \arg\max_{\kappa} p(\kappa \mid x^{(i)})$

IV. Top-k Lists Similarity Measures

In order to evaluate the accuracy of the predicted results with the actual, we employed a variety of measures that capture the similarity between two different top-k lists.

*A. OSim: D*enoted as *OSim(A,B)* [3] and measures the number of common elements of two sets A and B (each of size k):

$$OSim(A, B) = \frac{|A \cap B|}{k} \qquad (11)$$

B. KSim

This is a generalization of Kendal Tau distance measure [4], which measures the number of concordant pairs of objects in lists A, B:

$$KSim(A,B) = \frac{\left|\{u,v) : A', B' agree - on - order\}\right|}{\left|A \cup B\right|(\left|A \cup B\right| - 1)(1/2)} \quad (12)$$

where:

$$A' = A \cup (B - A)$$
$$B' = B \cup (A - B)$$

The penalty parameter that we used for Case 4 in [4] is 0.5.

C. nDCG

The third measure, *nDCG* (Normalized Discounted Cumulative Gain) [5], is defined as:

$$nDCG(A,B) = \frac{DCG}{IDCG} \quad (13)$$

$$DCG = CG(1) + \sum_{i=1}^{k} \frac{CG(i)}{\log_2(i)} \quad (14)$$

where $CG(i) = k - |i - j|$ and *i/j* is the real/predicted position of the web page.

IDCG is the ideal *DCG* where the two lists *A*, *B* are in complete agreement. In this case:

$$IDCG = k + \sum_{i=2}^{k} \frac{k}{\log_2(i)} \quad (15)$$

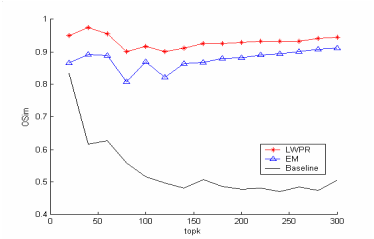

Figure 1. Top-k OSim scores

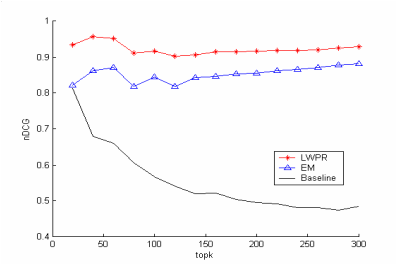

Figure 2. Top-k nDCG scores

D. Spearman Footrule

This measure is the Spearman Footrule distance for top-k lists. We use the generalization of Spearman Footrule distance as described in [4] with location parameter $L = k + 1$, for two *top-k* lists *A*, *B* of size *k* each one. Assuming that *A(i)/B(i)* returns the ranking position of page *i* in *top-k* list *A/B*:

$$F^{(L)}(A,B) = \sum_{i \in A \cup B} \left|A'(i) - B'(i)\right| \quad (16)$$

$A'(i) = A(i)$, if *i* belongs to *A* and $A'(i) = L$ otherwise. The $B'(i)$ function is defined similarly. Finally, the score of the predictions is given by:

$$score(A,B) = 1 - \frac{F^{(L)}(A,B)}{\max\{F^{(L)}(A,B)\}} \quad (17)$$

The maximum value of the Footrule distance in two top-k lists occurs when $A \cap B = \varnothing$, and in this case:

$$F^{(L)}(A,B) = k^2 + k \quad (18)$$

V. Experiments

A. Data set and preprocessing

The data set consists of approximately 1 million articles from the graph of Wikipedia, where we extracted the pagerank values and other statistics such as normalized pagerank, number of inlinks and number of outlinks of each page. The aforementioned data were extracted from the Wikipedia dump, available at: http://download.wikipedia.org/, for 23 consecutive months. During the preprocessing phase we extracted from the whole data set the pages/articles that received a ranking position between 1 and 300 in at least one of the 23 months. The final data set includes approximately 1500 pages and for each page we constructed a time sequence of racer values and each racer value at time t is related with a feature vector with the following variables: *1) The normalized pagerank of the article. 2) The pagerank value 3) The number of inlinks 4) The number of outlinks.*

In the generalized description of the algorithms in section III this feature vector at time *t* is denoted as $f^{(c)}$ with *c* = *t*.

VI. Experimental Methodology

In order to choose the parameters of our learning algorithms that maximize the accuracy of predictions we used a train-evaluate-test procedure in separate partitions of the data set.

A. Parameter Selection for Locally Weighted Polynomial Regression

The parameters that we had to choose for this model were the bandwidth parameter *r* of the weights $w^{(i)}$ in the error function "(1)", the degree of the polynomial function and the

training window. There are basically three approaches that we consider for the training window: **i**) choose a fixed month and start training our model up to the last month of the training set (**Fixed Mode**), **ii**) slide the training window each time we predict for the next month and retrain the model (**windowSlideAtBegin**). For example if we trained our model in months [1-20] then we predict for month 21, train in [2-21] and predict for month 22, etc. **iii**) slide window only at the end (**windowNoSlideAtBegin**). Another parameter that we also needed to specify was the starting month of training (**startMonth**).

The most promising results are obtained if we set the parameter **startMonth** = 8, r = 0.8, polynomial degree 6 and **windowSlideAtBegin**.

B. Parameter Selection for EM-Clustering

In this method, as far as the training window, we considered the **Fixed** mode, as it produces high similarity scores with low training complexity. However, we have experimented with the **startMonth** parameter and the polynomial degree as well.

In figures [1, 2, 3, 4] we can see the performance of LWPR and EM clustering, in the test set, in comparison to a *baseline predictor* which returns the most frequent ranking position from the historic data of each web page.

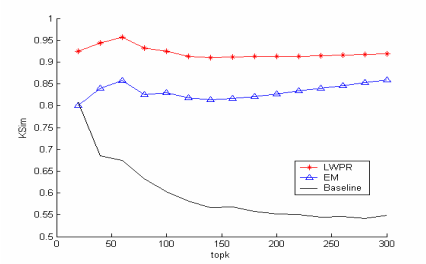

Figure 3. Top-k KSim scores

The parameters' values (**startMonth**=8, r=0.8, polynomial degree 6, **windowSlideAtBegin** for LWPR and **startMonth**=8, polynomial degree 6, 15 clusters with **Fixed** window mode for EM) are those that gave the best performance during the evaluation procedure. The above values were obtained empirically trying different set of combinations in the evaluation set. Overall the performance of the predictive framework is highly encouraging as the predictions accuracy ranges systematically between 80% and 98% while clearly outperforming baseline predictions.

VII. CONCLUSION

In this paper we introduced two regression algorithms capable of accurately predicting the rank of a graph entity comprised of multiple features which can change and evolve dynamically through time. Specifically, the first method that we employed is Locally Weighted Polynomial Regression which is initially presented in [2]. The second algorithm is a combination of the EM clustering with polynomial regression models. In our work we have evaluated and presented the efficiency of these methods in graph based environments, with multi feature nodes, that evolve in time. The prediction results show that using high order polynomial degrees, in both LWPR and EM Clustering, high similarity scores can be achieved.

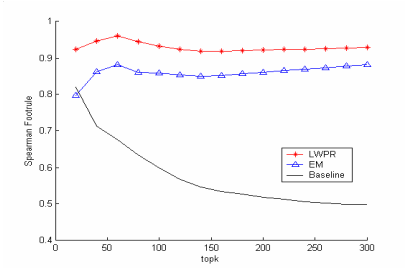

Figure 4. Top-k Spearman Footrule scores

REFERENCES.

[1] P. Zacharouli, M. Titsias and M. Vazirgiannis, Web Page Rank Prediction with PCA and EM Clustering, Workshop on Algorithms and Models for the Web graph, Feb. 2009.

[2] W. S. Cleveland, Robust Locally Weighted Regression and Smoothing Scatterplots, Journal of the American Statistical Association, 74, 368, 829-836, December, 1979

[3] T. H. Haveliwala, Topic-sensitive PageRank, In Proc. WWW, Honolulu, USA, May 2002.

[4] Fagin, R., Kumar, R., and Sivakumar, D. (2003). Comparing top k lists. SIAM Journal on Discrete Mathematics 17: 134–160.doi:10.1137/S0895480102412856

[5] K. J¨arvelin and J. Kek¨al¨ainen, Cumulated gain-based evaluation of IR techniques, TOIS,20(4):422–446, Oct. 2002.

[6] A. Vlachou, K. Berberich, and M. Vazirgiannis, Representing and quantifying rank-change for the Web graph, In Proc. WAW, Banff, Canada, Nov. 2006.

[7] Bishop, Christopher (2006). Pattern Recognition and Machine Learning. Berlin: Springer. ISBN 0-387-31073-8

[8] Scott Gaffney , Padhraic Smyth, Trajectory clustering with mixtures of regression models, Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, p.63-72, August 15-18, 1999, San Diego, California, United States [doi>10.1145/312129.312198]

[9] Scott Gaffney and Padhraic Smyth. Curve clustering with random effetcs regression mixtures. In C. M. Bishop and B.J. Frey, editors, Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics 2003.

[10] Lall U, Moon YI, Kwon HH, Bosworth K. 2006. Locally weighted polynomial regression: parameter choice and application to forecasts of the Great Salt Lake. Water Resources Research 42*:* W05422, doi:10.1029/2004WR003782

[11] Joshua O'Madadhain , Jon Hutchins , Padhraic Smyth, Prediction and ranking algorithms for event-based network data, ACM SIGKDD Explorations Newsletter, v.7 n.2, p.23-30, December 2005.

[12] Ruma Dutta, Anirban Kundu, Rana Dattagupta, Debajyoti Mukhopadhyay, An Approach to Web Page Prediction Using Markov Model and Web Page Ranking, Journal of Convergence Information Technology Volume 4, Number 4, December 2009.

[13] Debajyoti Mukhopadhyay, Priyanka Mishra, Dwaipayan Saha, Young-Chon Kim, A Dynamic Web Page Prediction Model Based on Access Patterns to Offer Better User Latency, arXiv:1102.0684v1

[14] Yong Zhen Guo , Kotagiri Ramamohanarao , Laurence A. F. Park, Personalized PageRank for Web Page Prediction Based on Access Time-Length and Frequency, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, p.687-690, November 02-05, 2007 [doi>10.1109/WI.2007.145].