Personalized User Profiling for Time-Dependent Recommendation of Structured Products

Richard Searle
Eccentric Data Limited
Botanic House, 100 Hills Road,
Cambridge, United Kingdom
richard@eccentricdata.com

Megan Bingham-Walker
Eccentric Data Limited
Botanic House, 100 Hills Road,
Cambridge, United Kingdom
megan@eccentricdata.com

ABSTRACT

The goal of a well-implemented recommendation system is to connect users with the items or content that appeals to them most. Scenarios where one or more elements of the user-item relationship are variable and where this variability is time-dependent represent a significant challenge to the design of efficient recommendation systems. By discussing the development of such a system in the context of a financial services application, the contribution of this paper is the disclosure of: 1) a novel modeling technique to effectively represent dynamic user preferences in an open domain space and 2) a sketching algorithm for efficient user profiling over a sparse, hyper-dynamic dataset.

CCS Concepts

•**Mathematics of computing** → **Graph algorithms;** •**Information systems** → **Network data models;** *Temporal data;* •**Theory of computation** → **Dynamic graph algorithms;** *Unsupervised learning and clustering;*

Keywords

Dynamic user profiling, recommendation systems, network models, community detection.

1. INTRODUCTION

Best-in-class recommendation systems are differentiated by the quality of their user preference profiles [6]. Creating user profiles is currently well understood, where there is a relatively stable inventory of products or content, where it is possible to gather detailed metadata on the characteristics of the inventory of items and where it is possible to amass significant data on user ratings [6]. In contrast, in this paper, we describe an efficient, structural, approach to incrementally learning clusters of associated users and products from high-dimensional, time-series data. We apply this technique to the personalized user profiling of the clients of a financial services company specializing in structured products.

Recsys '16 Boston, Massachusetts USA

ACM ISBN 123-4567-24-567/08/06...$15.00

DOI: 10.475/123_4

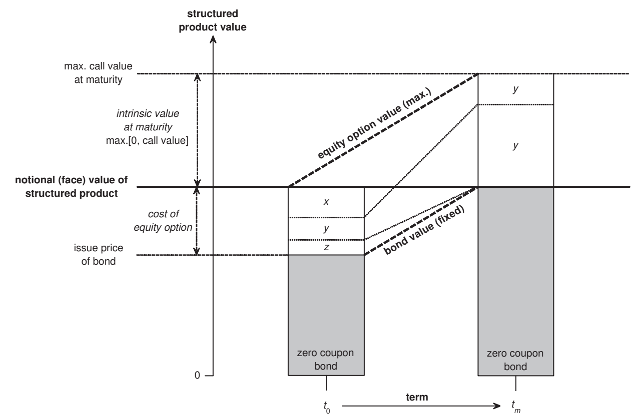

Figure 1: Illustration of a structured product comprising a zero coupon bond of fixed maturity value and three underlying assets: x, y, and z. At the time of purchase, t_0, the nominal (face) value of the structured product is that of the zero coupon bond at maturity, t_m. The investor waives the discount in the value of the zero coupon bond in order to participate in any gain in the call value of an arbitrary equity option at t_m. In the example shown, the value of asset z diminishes to zero and the intrinsic value of the structured product derives from the value of assets x and y only. Since the value of the zero coupon bond at maturity is equivalent to the purchase price of the structured product, in the event that the equity call option is worthless at t_m, the investor benefits from principal protection over the term of the investment.

A structured product is an investment product that comprises a basket of underlying financial instruments, for example, equities, debt issuance, commodities, currencies or a combination of globally traded securities. A detailed exposition of the various forms of structured product and the market for these complex investments is beyond the scope of this paper, however, an example of a structured product investment comprising a basket of underlying assets is shown in Figure 1.

Structured products are designed to incorporate highly customized risk-return objectives and, as such, cannot be recommended to users (or customers) in the sense that a movie or consumer product may be promoted to a particular individual. Instead, structured products can be offered to the market, where an informed investor may choose to purchase the structured product at the terms offered at a particular point in time. Thus, a timely structured product offer that approaches the optimal investment criteria of a specific user community represents a source of strategic competitive advantage to the vendor. It is the realization of this advantage that motivated the development of this novel system.

In designing a system to derive potential structured products that are appropriate to specific user community, two challenges were identified to the development of successful user profiling:

Firstly, the sparse hyper-dynamic nature of the product and user preferences that is generated by the global securities markets necessitated an automated, algorithmic ranking metric. User preferences are conditioned with respect to the characteristics of a particular structured product offer, whereas individual structured products are a function of their underlying components. This decoupling of user preferences from the attributes of the structured product proves problematic when trying to identify an affinity between discrete users and potential structured product offers using techniques such as latent factorization. This meant that an alternative modeling technique was needed to allow dynamic updates over the open structured product and user preference domain and to alleviate cold-start issues for new users.

Secondly, the volume and complexity of the example dataset that we discuss meant that it was necessary to balance accuracy with computational efficiency. We report the combination of a similarity measure (with a threshold parameter for constraint of the dynamical network model) and fast community detection as the basis for an effective sketching algorithm.

2. DEFINITION OF THE PROBLEM

We present the case of a dynamic user profiling application in the financial services sector, for which we have successfully deployed a scalable network model with near linear computational performance. Our client sought the capability to define structured products that are appropriate to the prevailing characteristics of specific service users and groups of users in both a reactive and pre-emptive capacity.

To facilitate development of the system, training data was provided in the form of time-series data for ∼61k anonymous user price requests that were received by the trading desk over a one-month sample period. In addition to responding to structured products offered to the market by the client, users can initiate a price request for a specific class of structured product with prescribed characteristics, e.g.: a given return, duration or principal value, or a given set of underlying assets or asset classes, etc. Price requests for structured products may be rejected, repeated, or traded (i.e. purchased) by a user and are time-dependent. Each price request and trading instruction is recorded by the client and to facilitate development of the system, training data was: the class of structured product requested, the status of the price request (i.e. whether the price request was converted to a traded structured product or was rejected by the user), the composition and valuation of the basket of underlying financial instruments forming the structured product offer, and metadata pertaining to the attributes of each underlying asset. Examination of the received data indicates, however, that there is a random relationship between a user's price request history and their trading history. Hence, personalized user profiling on the basis of this data posed several challenges to system design that have, hitherto, prevented accurate profiling of service users and their dynamic behavior:

1. The open nature of the user community, their governing preferences, and the latent factors affecting conversion of price requests to traded structured products.
2. The combinatorially explosive set of possible underlying asset combinations defined by global securities markets and the continuously variable underlying asset prices and structured product features (e.g. product term, principal protections, and call options over the underlying assets). In practice between 1 and 9 unique underlying assets formed the basis for structured products for which price requests were made during the sample period.
3. The sample dataset provided by the client included ∼1,450 underlying assets, which occured within the total number of price requests with a maximum frequency of 5.6%. Hence, there was a sparse association between underlying assets and both price requests and traded structured products.
4. The four-dimensional space defined by the attribute features of the individual users (U), structured product assets (A), and component underlying assets (a) is open and continuously variable in response to systemic and exogenous feedback over time. The incorporation of feedback, especially positive reinforcement, has been shown to be problematic when modeling evolutionary dynamical systems [15] and the schematic representation of the example context is shown in Figure 2.
5. In addition to variation in the user profile, as an outcome of their price request and trading history, existing structured product portfolio $\{A\}$, and environmental contingency; the client's assessment of the user is also conditioned by these factors and serves to influence the evaluation of potential recommendations.
6. The cold-start problem [13, 1, 17], whereby profiling of new users, without a known price request and trading history, must be established on the basis of rational priors.

To solve this problem, we introduce a novel approach to developing an undirected probabilistic graphical model. Rather than building a graphical representation of distributions where the graph expresses the conditional dependence structure between random variables, we sought to use a graphical structure as a framework to learn the contextual co-occurrence patterns that bring together item-to-item, item to-user, and user-to-user similarity within the data. This is what is being referred to in this paper as the single graphical model. The principal features of our approach are:

- A dynamical graphical model that incorporates an open domain of products and users and which allows for the graceful emergence and decay of relationships over time.

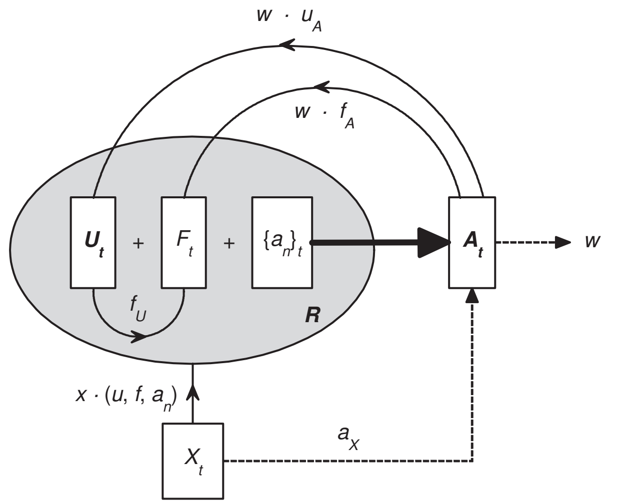

Figure 2: The problem context and recommendation space, $\boldsymbol{R}$, comprising the temporal state of the user profile $\boldsymbol{U}_t = \{\boldsymbol{u}|\boldsymbol{u} \in \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_i\}\}_t$, the client financial institution assessment of a user $\boldsymbol{F}_t = \{\boldsymbol{f}|\boldsymbol{f} \in \{\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_i\}\}_t$, and the basket of underlying assets $(\boldsymbol{a}_n)_t = \{\boldsymbol{a}|\boldsymbol{a} \in \{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_n\}\}_t$ comprising the call option of a structured product, $\boldsymbol{A}_t$, with weighted systemic feedback, $\boldsymbol{w}$, and exogenous feedback, $\boldsymbol{x}$, from the temporal environment, $\boldsymbol{X}_t$.

- A single network model with embedded contextual information for dynamical analysis and multi-dimensional community detection.
- Dynamic identification of communities of users with common interests for making timely and appropriate personalized recommendations, according to their momentary and projected relevance.
- Recommendation of products where the composition or features of the product is variable over time (i.e. structured product assets and effective investment values).
- Efficient performance over high-dimensional, sparse, datasets, with near linear scalability.
- Incremental representation learning with/without supervision.

3. METHODOLOGY

In addressing the problem application we established the systemic logic shown in algorithm 1 and algorithm 2. We begin by defining a model representation of the application domain at a current time, index t_α, that is updated for all subsequent times, t_β, for which data is received. All domain entities and their bipartite relationships are encoded in the graphical model by means of an undirected edge. The resultant dynamic model is parameterized in order to restrict the density of the graphical model and maintain computability using a threshold parameter, p. User profiling is achieved by superposition of all communities to which their price request and trading activity is affiliated under the parametric constraints. Thus, user profiling (and/or profiling of any discrete entity or class within the domain) can be achieved across any one of more complementary dimensions within the model. This capability enables users to be associated with other users, different asset classes, different product values, and any other dimension with which a user is associated.

Algorithm 1 Building Graphical Model

```
procedure CreateGraph(G)
    load data for time t_β
    for price request in data do
        update user-structured product edge in G
        update structured product-asset edge in G
        update user-asset edge in G
    update threshold parameter, p
    for all entity pairs do
        compute, s_ij ≥ p
    constrain graph: G ↦ G_p
    for evaluate communities do
        algorithm 2
```

3.1 Model Structure

To enable dynamic user profiling for the described structured products application we define a network model in the form of an undirected probabilistic graphical model, G. The graphical model is composed of a set of vertices, $\upsilon \in V$, where each individual vertex, υ_i, represents a unique entity within the application domain; e.g. a user, a structured product, or an underlying asset. A vertex is characterized by a set of features, θ_k that are specific to that entity, such that: $V = \{\upsilon \mid \upsilon_i = \{\theta_1, \theta_2, ..., \theta_k\}_i\}$. Association of domain entities is represented by a single undirected edge, $e \in E$, where e_{ij} defines the relationship between two entities υ_i and υ_j.

The systemic knowledge of the application problem space is only partial, since the latent features that condition user preferences, pricing, and trading decisions are unknown. We acknowledge this restriction within the model by considering only those features, $(\theta_k)_i$, that the system has observed. This constraint enhances the computational efficiency of the model and is consistent with research that questions the validity of recommendations derived using implicit feedback models with latent factorization [12].

3.2 Graph Morphology

A key characteristic of the network approach is the continuous dynamic reconfiguration of the graphical model structure under parametric constraints. The complexity introduced by the variable composition and value of the structured product assets precludes the use of a static network representation, as is common in social network analysis and other fields [8].

Unique vertices within the graphical model are associated using a similarity distance measure that is a function of the observed interaction between the domain entities. Symmetric associations are made according to the triadic relationship model shown in Figure 3.

The relational model in Figure 3 may be extended by any degree, in response to expansion of the domain ontology. We employ the Jaccard similarity measure [9, 10] as a characteristic attribute of the edge e_{ij}, whereby:

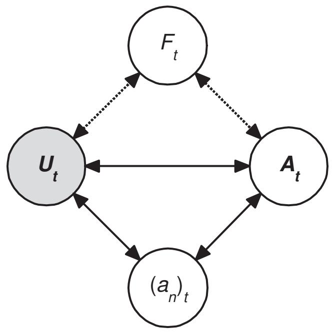

Figure 3: The triadic relationship model of domain entities (with implicit client interaction) represented by the structure of the undirected graphical model *G*.

$$s_{ij} = \frac{|\upsilon_i \cap \upsilon_j|}{|\upsilon_i| + \upsilon_j| - |\upsilon_i \cap \upsilon_j|} \tag{1}$$

$$0 \leq s_{ij} \leq 1$$

The Jaccard coefficient has been successfully deployed over a variety of large-scale networks [8] to identify the latent structure of high-dimensional domains with sparse interaction.

We compute the measure s_{ij}, subject to the satisfaction of two systemic constraints. Firstly, we impose a threshold parameter, p, which conditions the consideration of entity associations when deriving a user profile. Network edges are masked within the model if the value of s_{ij} is less than the threshold p, thereby conditioning the structural relationships represented by the model. Where edges are temporarily masked, the graphical model attributes continue to be updated without any loss of information. Secondly, a simple transposition of the formula in equation 1 is used to evaluate whether the cardinality of the edge vertices is sufficient for s_{ij} to exceed the current value of p. Typically, the value of the masking parameter is arbitrary, however, the threshold may be updated dynamically as a function of the size of the network or the cardinality of the vertices.

Masking of the edges with similarity coefficients below the threshold value imbues the model with a dynamic morphology that is independent of any variation in the composition of the domain. Thus, addition of a new user, product, or underlying asset will only change the model structure if a statistically relevant association emerges within the model.

3.3 Community Detection

Using the dynamic network model as a basis, user profiling is undertaken by fast community detection, augmented by supplementary consensus clustering following the method described by Lancichinetti and Fortunato [11]. User profiles are constructed by fast approximation of network communities over the prevailing graphical model structure using the *Louvain* algorithm [2]. Several studies involving problems of community detection in large-scale social networks [16, 8] have successfully employed this method to efficiently partition graphical models with low latency and good approximation [11]. The computational efficiency of *Louvain* community detection is derived from the reductive greedy optimization of partitions over the graphical model using the modularity parameter, Q [14], where for the population of communities C:

$$Q = \sum_{n \geq 1} P(e_{ij} \in C_k) - P(e_r \in C_k)^2 \tag{2}$$

$P(e_{ij} \in C_k)$ = probability that edge e_{ij} has both endpoints in C_k.

$P(e_r \in C_k)$ = probability that a random edge e_r has ≥ 1 endpoint in C_k.

This procedure is expressed in algorithm 2 below:

Algorithm 2 Find User Communities

procedure FINDCOMMUNITIES(C)
  for G **do**
    find C_u user communities:
    Louvain: $\triangle(s_{ij})_t$
    find C_{prod} product communities:
    Louvain: $\triangle(C_{u \mapsto prod})_t$
  for communities C **do**
    evaluate consensus: $C_{u \mapsto prod}$
  for evaluate dynamics $t_\alpha \to t_\beta$ **do**
    Constituent: $\triangle(s_{ij})_t$
    Community: $\triangle(C_{u \mapsto prod})_t$
  for top-k target communities **do**
    compute rank C_k
  for top-k user-structured product pairs **do**
    compute $prod_k, \forall u \forall prod \in C_k$
  present data to client system
  for edges e_{ij} in G **do**
    incorporate feedback w

As the edge characteristic s_{ij} varies in proportion to the threshold parameter p, the membership of a graph entity within a specific community partition may change through the gradual emergence and decay of the Jaccard similarity measure, as shown in Figure 4:

Prior research has identified limitations in the modularity optimization method employed by the *Louvain* algorithm [4, 7]. Specifically, communities that may yield important information about the relatedness of users, products, or underlying assets may be lost as a consequence of proportional scaling issues in massive graphs [4]. To overcome this problem we use consensus clustering [11], which further resolves community boundaries by concentrating selection on co-clustered vertices. The induced consensus over community distributions represents a temporal snapshot from which tailored recommendations can be generated for the co-clustered model entities.

An important consideration in partitioning the model is the stability (or persistence) of the network communities over time and their intrinsic association of entities. Greene et al. [8] and Lancinchinetti and Fortunato [11] demonstrate the further utility of the Jaccard similarity measure in using to associate the communities detected by consensus co-clustering. By ranking communities according to their significance with respect to; 1) the dynamic stability of their constituent entities, or 2) the persistence of their identity over time [8] it is possible to evaluate a user profile according to its momentary state or dynamic trajectory. When com-

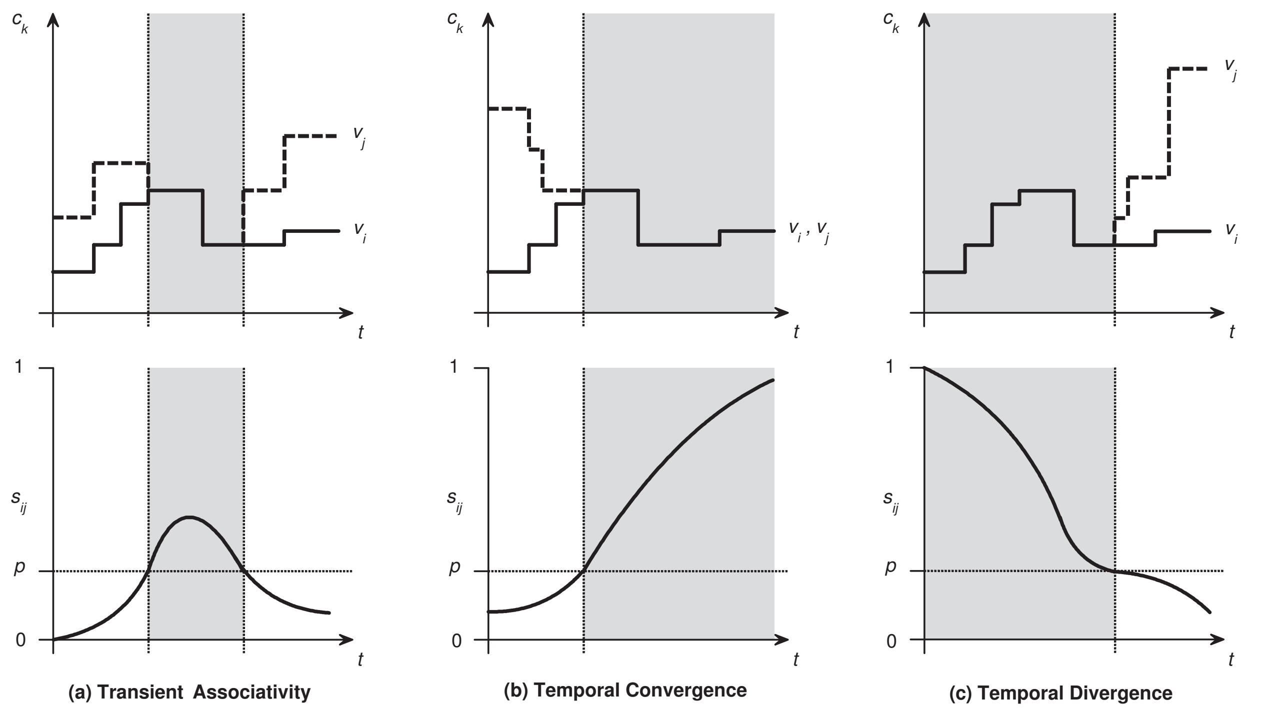

Figure 4: Schematic illustration of the temporal morphology of the dynamic network model by community affiliation of constituent vertices: a) transient association of vertices υ_i, and υ_j, through the temporal emergence and decay of the pair-wise similarity measure s_{ij}; b) temporal convergence of vertices υ_i, and υ_j, such that $s_{ij} \geq p$; c) temporal divergence of vertices υ_i, and υ_j, such that $s_{ij} < p$. Shaded areas indicate the condition: $e_{ij} \in C_k$.

bined with dynamic trend information, $\triangle s_{ij}$, for the edges that are intrinsic to a particular community, the latent dynamics of user behavior are made available to the process of recommendation without recourse to implicit factorization.

By comparing the temporal sequence of communities [11] defined by the system it is possible to detect changes in individual user behavior and early indicators of trends within the investment market. This capability enables the client organization to initiate a proactive response to market conditions by offering new products, directed towards a particular community of users, in a timely fashion. It is notable that, in the domain of structured products, false positive indicators that suggest structured products that are not taken up by the projected user do not carry the same downside risks as false negative indicators that suggest a user will reject a structured product that is offered. Hence, dynamic identification of positive user-product relations enables the client organization to participate in the market effectively by ensuring structured products are available at the moment conditions consistent with a user profile emerge.

4. DISCUSSION

The training dataset simulated a continuous stream of global price-request data for the sample period and was presented as a series of time-indexed JSON files. Due to the low volume of price-requests received on weekends, the number of samples per day was in the range: $1 \leq |samples|_{day} \leq 3,580 (x \cong 1,951)$ with a total of 60,479 samples covering 31 consecutive days.

We implemented the described dynamic user profiling and recommendation system using a commercially available laptop computer, without the use of parallel computing or any other form of system optimization. The host computer included: a 2.8GHz Dual-core Intel® Core™ i7 processor; a 16GB 1600MHz DDR3L SDRAM memory module; and, a 512GB flash storage device that was used to store the sample dataset and the persisted undirected graphical model, G. Iterative learning and reconfiguration of the network model was undertaken as an end-of-day batch job, consistent with the operational procedure of the client.

The computed polymorphic domain model for the training dataset contained 61,925 vertices and 154,088 edges and the batch runtime was found to be consistent and approximated to the polylogarithmic class $O\ (\log\ n/2)^2)$, where $n \propto |V|$. Figure 5 illustrates the relationship between system runtime and the size of the graphical model.

Community detection was completed in less than one second for each model configuration using the *Louvain* algorithm of Blondel et al. [2]. The efficient performance of the employed community detection method and the potential for parallel processing of the graphical model in real-time demonstrates the scalability of the system to large-scale applications with higher volumes of input data or a base network model of higher rank. We did not apply parallel processing techniques within this study.

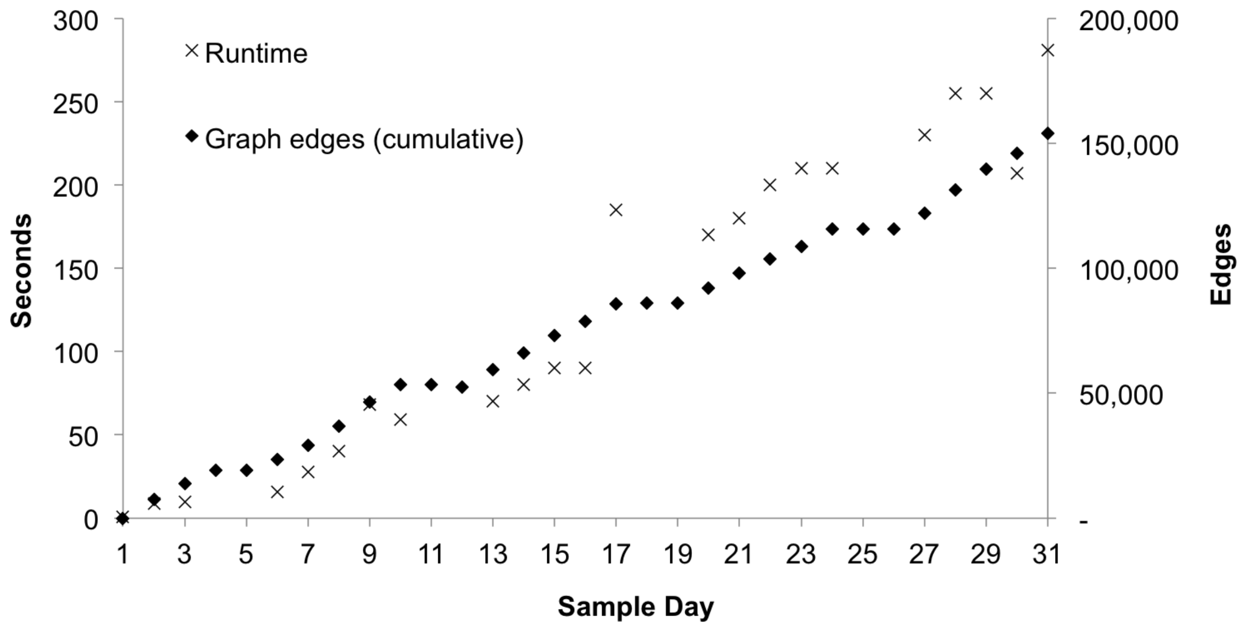

Figure 5: Cold-start system performance for daily learning and reconfiguration of the dynamical network model (end-of-day batch update).

5. RECOMMENDATION

Having identified the latent community structure within the temporal model, the static and dynamic attributes of the graphical model can be used to formulate a temporal user profile and extract item-item, item-user, or user-user relationships so as to derive an appropriate structured product offer. To compose a structured product for pricing, the recommended model attributes are fed into the trading system of the client, where a proprietary pricing algorithm is used to derive an structured product offer that is appropriate to the temporal characteristics of the user.

The recommendation algorithm is based on a calculated mean-path-rank of the community for a given set of selection criteria. The mean-path-rank algorithm selects the most relevant basket of underlying assets from which a structured product may be offered, according to the structured product criteria and effective market pricing of the assets. The mean-path is a function of the similarity relation $(s_{ij})_{a\leftrightarrow a}$ that exists between potential assets and the relation $(s_{ij})_{u\leftrightarrow a}$ that maps a user to each asset comprising the potential structured product. Reinforcement learning is incorporated by means of the feedback weighting, w, shown in Figure 2. The mean-path-rank algorithm is naïve to the ground truth of the prior community structure yet, under the methodology set out by Lancinchinetti and Fortunato [11], we found good correspondence between trading patterns and associated price request data within the training dataset. This is illustrated by Figure 6. The correspondence between the derived community structure and actual trading performance indicates that the structured products offered to users based on their community membership are consistent with user demand. We, therefore, assert a positive potential trading impact with the new user insights furnished by the system but are unable to report actual performance gains within this paper as a result of commercial undertakings with our client.

In practice, a key characteristic of the system has proved to be the integration of graceful emergence and decay of the similarity measure s_{ij}. Even where prior knowledge of an application domain is assumed to be complete, the concepts that need to be learned may change in their contextual relationship to one another over time as a result of concept drift [5]. It is this property of the dynamic network model, illustrated schematically in Figure 6, that enables consistent recommendation accuracy in response to different market conditions and changes in the user profile (e.g. portfolio value, price request history, latent investment criteria).

6. CONCLUSIONS

The idea of using a graphical approach to recommendations is not new. There have been significant advances in search and recommendation in recent years as a result of the introduction of large-scale knowledge graphs. However, these are typically static representations, which add additional links as new facts become available but “the ground truth values associated with the links [are] considered time invariant” [3].

In addressing the reported financial services application on behalf of our client we found that established recommendation methods, premised on static network representations

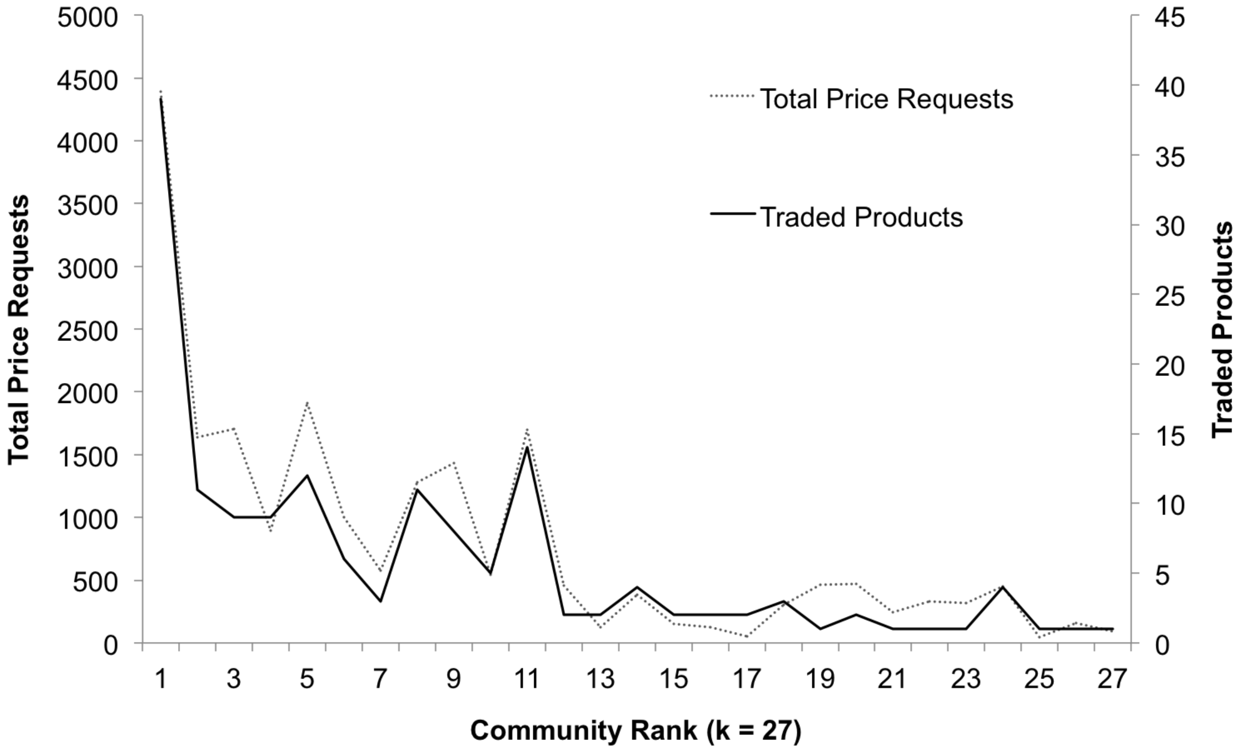

Figure 6: Correspondence of identified community ranking, C_k, with independent price request and traded structured product data for the dynamic network model (day 31, full rank graph: $|\boldsymbol{V}|$ =61,925 and $|\boldsymbol{E}|$ =154,088).

of a domain [8], were incompatible with the sparse hyper-dynamic nature of the problem space. We suggest that the use of parametric thresholds for restriction of the model edge density - in combination with a fast, scalable, community detection algorithm - is a prerequisite for the dynamic user profiling and recommendation systems demanded by large-scale, complex, and time-dependent industrial applications.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, October 2008.

[3] C. Esteban, V. Tresp, Y. Yang, S. Baier, and D. Krompaß. Predicting the co-evolution of event and knowledge graphs. *arXiv preprint arXiv:1512.06900*, 2015.

[4] S. Fortunato and M. Barthèlemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[6] C. A. Gomez-Uribe and N. Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, January 2016.

[7] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[8] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *Advances in social networks analysis and mining (ASONAM), 2010 international conference on*, pages 176–183, Los Alamitos, August 2010. IEEE.

[9] P. Jaccard. Ètude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Sociètè Vaudoise des Sciences Naturelles*, 37(142):547–579, 1901.

[10] P. Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, February 1912.

[11] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific reports*, 2(336):1–7, March 2012.

[12] D. Liang, L. Charlin, J. McInerney, and D. M. Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 951–961, Geneva, 2016. International World Wide Web Conferences Steering Committee.

[13] D. Maltz and K. Ehrlich. Pointing the way: active collaborative filtering. In *Proceedings of the SIGCHI conference on Human factors in computing systems*,

pages 202–209. ACM Press/Addison-Wesley Publishing Co., 1995.

[14] M. E. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, March 2004.

[15] C. Pahl-Wostl. *The dynamic nature of ecosystems: chaos and order entwined*. John Wiley & Sons Incorporated, Chichester, 1995.

[16] T. Raeder and N. V. Chawla. Market basket analysis with networks. *Social network analysis and mining*, 1(2):97–113, April 2011.

[17] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.