Graph Embeddings for Movie Visualization and Recommendation

Michail Vlachos
IBM Research - Zurich, Switzerland

Daniel Svonava
Slovak University of Technology, Slovakia

ABSTRACT

In this work we showcase how graph-embeddings can be used as a movie visualization and recommendation interface. The proposed low-dimensional embedding carefully preserves both local and global graph connectivity structure. The approach additionally offers: a) recommendations based on a pivot movie, b) interactive deep graph exploration of the movie connectivity graph, c) automatic movie trailer retrieval.

1. INTRODUCTION

As we are moving gradually from the era of information to the era of recommendation, interactive interfaces that engage the users and let them easily discover the data of interest will be of increasing importance. In this work we explore how graph-embedding techniques can be used as the basis of an interactive recommendation engine for movies.

Several recommendation systems have appeared in the literature in the recent years for recommending videos [5, 9] or movies [2, 6, 8, 10]. These, however, rarely focus on visually driven interfaces. In our scenario, we use a movie-actor database as the underlying graph structure. We use textual features to describe the movie objects. Given a selected pivot movie the system can retrieve a set of similar movies which are portrayed and clustered on two dimensions. Our method presents a novel way of capturing both neighborhood and cluster structure. Neighborhood information is preserved by retaining the Minimum Spanning Tree (MST) structure on two-dimensions. This also partially preserves the global graph structure, as the MST represents the dataset 'backbone'. In addition to the neighborhood structure, the method also retains the cluster structure which can be visualized at different granularities. Finally, the proposed mapping can accommodate both metric and non-metric distance functions.

Data-embedding techniques have been extensively used for visualizing high-dimensional data. Examples include the Bourgain embedding [3], FastMap [7], BoostMap [1] and ISOMAP [12]. The goal of those projection methods is to preserve all distances approximately, while our approach preserves a subset of distances (spanning-tree distances) *exactly*.

We use the proposed graph embedding as the entry point for a movie recommendation interface. Our methodology allows the exploratory visualization of the movie graph space and incorporates additional capabilities, such a filtering of the graph based on various criteria, real-time graph exploration and automatic retrieval of the movie trailer from the Internet.

2. DESCRIPTION OF OUR APPROACH

The proposed method combines the visual simplicity and comprehensibility of the minimum spanning tree (MST) with the grouping properties of hierarchical clustering approaches (dendrograms). Given pairwise distances describing the relationship of high-dimensional objects, the objective is to preserve a subset of important distances as well as possible on 2D, while at the same time accurately conveying the cluster information.

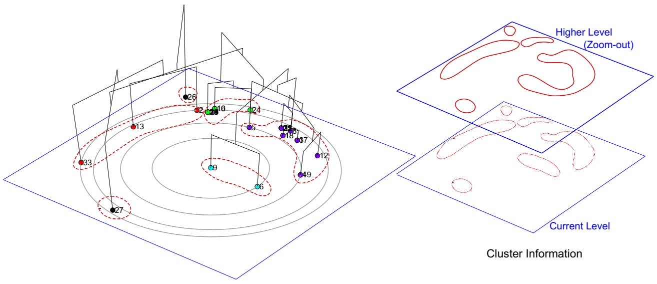


Figure 1. Conceptual illustration of our approach. The spanning tree structure is preserved and projected onto a 2D plane, while cluster structure is overlayed in the third dimension. 'Cutting' the dendrogram at a certain level projects the cluster structure onto 2D.

Our approach works as follows. First, we construct a Minimum Spanning Tree (MST) layout on the 2D plane in such a way that all distances to a user-selected pivot point and the neighborhood distances on the MST are exactly preserved. This construction carefully considers how to best portray the original object relationships for either metric or non-metric distance functions. Secondly, the dendrogram cluster hierarchy is constructed so that it can be positioned exactly *on top* of the MST mapping of objects. The cluster hierarchy can be frozen at any resolution level (tomographic

Paper presented at the Workshop on Interfaces for Recommender Systems 2012, in conjunction with the 6th ACM conference on Recommender Systems. Copyright @ 2012 for the individual papers is held by the papers' authors. This volume is published and copyrighted by its editors. Interface@RecSys'12, September 13, 2012, Dublin, Ireland
.

	Visualization Method	Trailer	Filters	Special Features
netflix.com	Tables	Yes	Limited	Very large user-base
jinni.com	Linear Treemap	No	Yes	Semantic search
IMDB.com	Icon Tiles	Yes	No	Comprehensive database
MovieLens.org	Tables	No	Yes	Recommendations through initial test
Our method	Minimum Spanning Dendrograms	Yes	Rating/Genre/Year	Clustering/Deep Graph Exploration

Table 1. Overview of features for several extant movie recommender systems

view) to convey the multi-granular clusters that are formed. This concept is elucidated in Figure 1: objects are properly mapped on the 2D plane whereas on the third dimension the hierarchy of the clustering structure is portrayed. Naturally, this is only a conceptual illustration of our approach. In practice, cluster information is also projected onto two dimensions, e.g., by properly coloring the nodes belonging to the same cluster. Therefore, by 'cutting' the dendrogram derived on a user-defined level, clusters on 2D can be formed, expanded and contracted appropriately, as the user drills up or down on the cluster hierarchy.

The work presented here constitutes an demo prototype of hhe visualization technique presented in [11]. Here we explore in more detail how the proposed high-dimensional data embedding methodology can be used as the interface of a movie search engine. In Table 1 we present briefly the differences of our approach with respect to prevalent movie search and recommendation systems.

2.1 Neighborhood Preservation

We will first explain how to capture on two dimensions the relationship between a set of high-dimensional objects. As not all pairwise distances can be retained on two dimensions we choose to maintain, as well as possible, the spanning tree distances which partially capture the local relationships and also record information about the general global structure [11].

We begin by constructing the spanning tree on the original high-dimensional objects. One object is selected as pivot and mapped in the center of the 2D plane coordinate system. By traversing the spanning tree, objects are positioned on the 2D plane by triangulating the distances to two objects: the pivot object and the neighboring point previously mapped on the spanning tree.

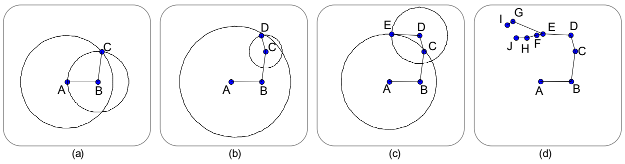


Figure 2. Two dimensional mapping of MST of objects

Example: Suppose the first two points (A and B) of the MST have already been mapped, as shown in Fig. 2(a). Let's assume that the second distance preserved per object is the distance to a reference point, which in our case is the first point. The third point is mapped at the intersection of circles centered at the reference points. The circles are centered at A and B with radii of $d(A, C)$ – the distance between points A and C– and $d(B, C)$, respectively. Owing to the triangle inequality, the circles either intersect in two positions or are tangent. Any position on the intersection of the circles will retain the original distances towards the two reference points. The position of point C is shown in Fig. 2(a). The fourth point is mapped at the intersection of the circles centered at A and C (Fig. 2(b)) and the fifth point is mapped similarly (Fig. 2(c)). The process continues until all the points of the ST are positioned on the 2D plane and the final result is shown in Fig. 2 (d).

The mapping technique presented will retain *exactly* the distances between all points and the pivot sequence, and also between the nodes that lie at the edges of the spanning tree. This creates a powerful visualization technique that not only allows to preserve nearest neighbor distances (local structure), but in addition retains distances with respect to a single reference point, providing the option of a global data view using that object as a pivot.

Layout optimization using simulated annealing: For reaching maximum visual clarity we try to minimize the number of intersected graph edges. Recall that when triangulating the position of a third point to its neighbor and the pivot, the algorithm proceeds by identifying the intersection between two circles. One can readily see this in Figure 3; there are two positions in which a newly mapped point can be placed.

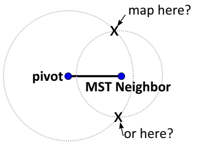


Figure 3. Selecting which of the two positions a new point is mapped to

We employ a probabilistic global optimization technique based on *simulated annealing (SA)* [4] that intelligently selects which of the two mapping positions to use, so as to minimize the number of crossed edges. SA is an effective optimization method when the search space is discrete, which is exactly the situation we face. Our experiments on the movie graph database suggest that the simulated annealing process is very effective in reaching an improved layout.

Using Non-metric Distances: When the underlying distance measure obeys the triangle inequality the circles around the reference points are guaranteed to intersect. However, many widely used distance functions (e.g., dynamic warping, longest common subsequence) violate the triangle inequality, and thus the corresponding reference circles may not necessarily intersect. In such cases, one needs to identify the position where to place an object with respect to the two circles, in such a way that the object is mapped as close as possible to the circumference of both circles. We need to identify the locus of points that minimize the sum of distances to the perimeters of two circles. One can show that the desired locus always lies on the line connecting the centers of the two circles. An example is shown in Fig. 4.

2.2 Cluster Preservation

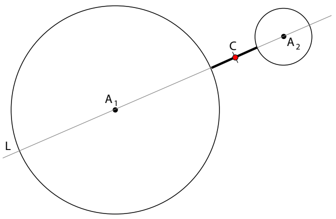


Figure 4. Discovering the mapping point for non-metric distances if the circles do not intersect.

Now we turn our attention to capturing and conveying the cluster information on 2D. Recall that the input for the algorithm is a matrix of pairwise distances. Based on the pairwise distances given, one can build a hierarchical dendrogram. The dendrogram construction is based on a single linkage approach that merges *closest* singleton objects and clusters.

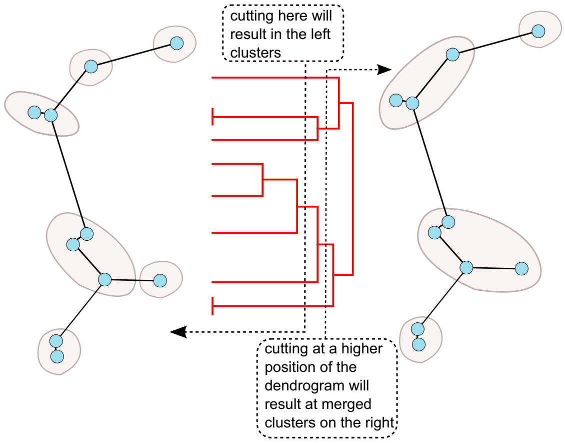

cutting here will result in the left clusters
cutting at a higher position of the dendrogram will result at merged clusters on the right

Figure 5. Cluster information conveyed by variable thresholds on a dendrogram information, thus imposing 'zoom-in' and 'zoom-out' in the cluster structure.

Given the minimum spanning tree, the clustering process can be sped up, because the merging order of the MST algorithm is the same as the merging order of the single linkage hierarchical clustering approach. In addition, this reflects the fact that one can achieve an *exact registration* of the constructed hierarchy on top of the MST-mapped points, because the clustering order is the same as the order crystallized in the spanning-tree mapping. This can be verified in Fig. 1 where the single linkage dendrogram is positioned exactly on top of the spanning-tree mapping.

The above observation combined with the hierarchical cluster information provides the capacity for multi-granular cluster views on 2D by interactively setting variable threshold levels on the resulting dendrogram. In this way, a 'tomographic view' of the clusters with formation of variable size clusters is possible. The concept is illustrated in Fig. 5: by cutting the dendrogram at a lower threshold, six clusters are created on the left. Imposing a higher threshold, clusters are merged progressively as shown on the right side. Our prototype implementation conveys cluster information by coloring the node perimeters and the connected edges.

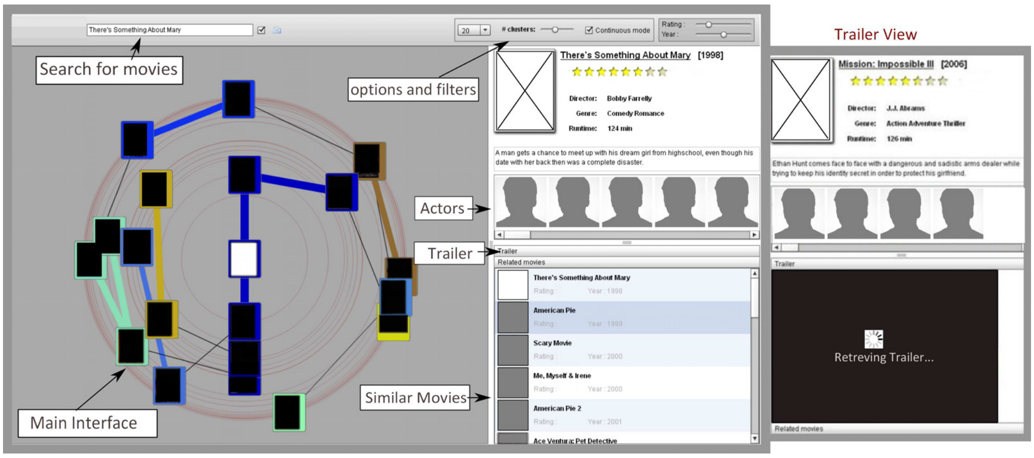
3. GRAPHICAL INTERFACE

On top of the proposed visualization methodology we have built a movie recommendation engine that allows the interactive exploration of a large movie graph. Our sample database consists of 125,000 movies and 955,000 actors. We augment the information on each movie by attaching additional unstructured information from the web. So, each movie is described by a *bag-of-words* pertinent to the genre, actors, director(s), language, and a set of keywords relevant to the plot. To evaluate movie similarities we follow an IR-driven approach by considering the cosine similarity between the bag-of-words. More complex functions can be also accommodated, however this approach already gave very satisfactory and intuitive results. So, similarity between movies is based solely on the content of the movie rather than on any collaborative features (or ratings). We follow this path in an effort to introduce a factor of *serendipity* in the recommendation process. Like this, more obscure movies can appear in the recommendation if they share similar content (e.g. plot or mood) with the one that the user selected. In general, we have noticed that movie recommender systems that consider collaborative features (e.g. users that have rated highly movie A have also rated highly movie B) tend to have a strong bias toward blockbuster movies that most users (typically) have already watched.

GUI and Functionalities: The data visualization is accommodated through a graphical web interface, shown in Figure 6. The interface allows the user to search for a movie, and subsequently displays the proposed visualization graph, placing the movie selected as the center (pivot) object. The user can then easily identify other relevant movies, with the option to retrieve detailed information about the movie, such as participating actors or the movie plot. The user can modify the number of displayed movie clusters or even watch the movie trailer. The application allows the exploration of both sides of the movie-actor bipartite graph: either by deeper exploring of the movie graph (by clicking on a movie) or by searching/filtering for movies of a particular actor (by clicking on an actor's image). Additional filtering functionalities include: a) Filtering by rating, e.g., when interested in retrieving movies with a rating $> Y$. b) Filtering by year, when the user is interested only in recent movie releases.

Below we provide specific examples of our mapping, highlighting its ability to be used as an effective and interactive movie recommendation system. Number of clusters can be interactively modified and clusters are conveyed using varying border and edge colors.

Examples: Selecting 'The Titanic' as pivot movie produces the 2D mapping shown on the left-hand side of Fig. 7. The user can navigate through the graph and identify similar movies. Clustered with Titanic are the movies: 'Titanic (1953)', 'Poseidon' and 'Shakespeare in Love'. Another cluster displayed, shown in light green, includes movies like: 'The Notebook', 'Atonement', 'Purple Rain'; all romantic drama movies. Similarly, selecting 'Star Wars' as pivot movie correctly packs closely the remaining Star Wars movies (Fig. 7, right-hand side). An adjacent cluster includes parodies of Star Wars, like 'Spaceballs' or 'Thumb Wars'. Other related movies include adventure films like the 'Lord of the Rings' trilogy, or sci-fi action thrillers like 'Aliens' and 'Star Trek'. Additional illustrative examples can be found in the provided video.



There's Something About Mary ☑

20 ▾ # clusters: ☑ Continuous mode Rating : Year :

There's Something About Mary [1998]

Director: Bobby Farrelly

Genre: Comedy Romance

Runtime: 124 min

A man gets a chance to meet up with his dream girl from highschool, even though his date with her back then was a complete disaster.

Trailer

Related movies

There's Something About Mary	Rating : Year : 1998
American Pie	Rating : Year : 1999
Scary Movie	Rating : Year : 2000
Me, Myself & Irene	Rating : Year : 2000
American Pie 2	Rating : Year : 2001
Ace Ventura: Pet Detective	

Mission: Impossible III [2006]

Director: J.J. Abrams

Genre: Action Adventure Thriller

Runtime: 126 min

Ethan Hunt comes face to face with a dangerous and sadistic arms dealer while trying to keep his identity secret in order to protect his girlfriend.

Trailer

Retreving Trailer...

Related movies

Figure 6. The interface of the Movie Recommendation web application

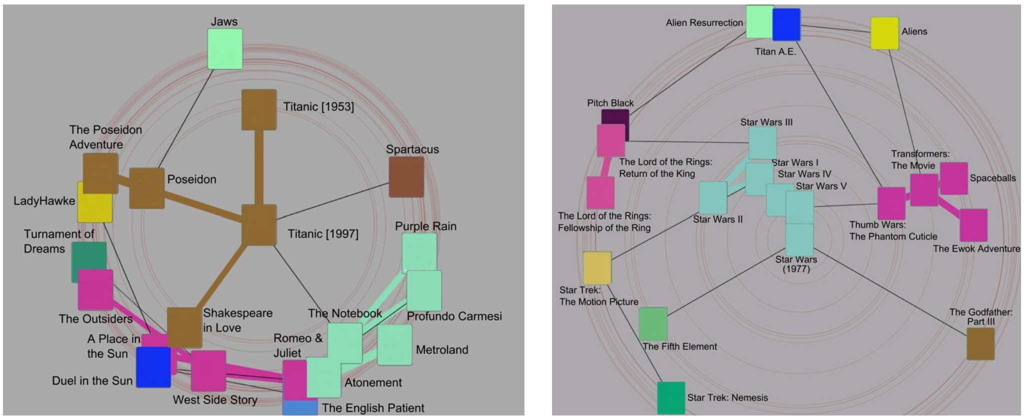


Figure 7. Using the proposed mapping on the movie graph. *Left:* Pivot movie is 'Titanic'. *Right:* 'Star Wars' selected as pivot

Benefits of simulated annealing (SA): The proposed SA component probes multiple node placement configurations, and picks the one that minimizes the number of intersected edges. To measure its effectiveness we select 1000 movies at random and retrieve the k nearest neighbors. Table 2 reports the median number of edge intersections with and without the SA component. We observe that the SA implementation significantly reduces the edge intersections and hence the screen clutter, providing a more intelligent node placement. Note, that both variations provide the same distance preservation with respect to the pivot and the MST distances.

Table 2. Node placement using simulated annealing (SA) significantly reduces the number of edge intersections

	Graph edge intersections	
	Without SA	With SA
k=20	36	12
k=30	108	48
k=40	222	112
k=50	346	212

References

[1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap: An Embedding Method for Efficient Nearest Neighbor Retrieval. In *IEEE Trans. Pattern Anal. Mach. Intell. 30(1)*, pages 89–104, 2008.

[2] J. Bennett and S. Lanning. The Netflix Prize. In *Proc. of KDD Cup and Workshop*, 2007.

[3] J. Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. In *Israel Journal of Mathematics, 52*, pages 46–52, 1985.

[4] V. Cerny. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. In *J. of Opt. Theory and Applications 45*, pages 41–51, 1985.

[5] J. Davidson et al. The YouTube Video Recommendation System. In *Proc. of ACM Recommender Systems*, pages 293–296, 2010.

[6] E. A. Eyjolfsdottir, G. Tilak, and N. Li. MovieGEN: A Movie Recommendation System. In *UC Santa Barbara: Technical Report*, 2010.

[7] C. Faloutsos and K. I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. of SIGMOD*, 1995.

[8] F. Gruvstad, N. Gupta, and S. Agrawal. Shiniphy - Visual Data Mining of movie recommendations. In *Stanford University: Technical Report*, 2009.

[9] T. Mei, B. Yang, X.-S. Hua, L. Yang, S.-Q. Yang, and S. Li. VideoReach: an online video recommendation system. In *Proc. of SIGIR*, pages 767–768, 2007.

[10] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. In *Proc. of IUI*, pages 263–266, 2003.

[11] D. Svonava and M. Vlachos. Graph Visualization using Minimum Spanning Dendrograms. In *IEEE International Conference on Data Mining*, 2010.

[12] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, pages 2319–2323, 2000.