Cache Content-Selection Policies for Streaming Video Services

Stefan Dernbach[1], Nina Taft[2], Jim Kurose[1], Udi Weinsberg[3], Christophe Diot[4], Azin Ashkan[4]

[1]University of Massachusetts, [2]Google, [3]Facebook, [4]Technicolor

***Abstract*—The majority of internet traffic is now dominated by streamed video content. As video quality continues to increase, the strain that streaming traffic places on the network infrastructure also increases. Caching content closer to users, e.g., using Content Distribution Networks, is a common solution to reduce the load on the network. A simple approach to selecting what to put in regional caches is to put the videos that are most popular globally across the entire customer base. However, this approach ignores distinct regional taste. In this paper we explore the question of how a video content provider could go about determining whether or not they should use a cache filling policy based solely upon global popularity or take into account regional tastes as well. We propose a model that captures the overlap between inter-regional and intra-regional preferences. We focus on movie content and derive a synthetic model that captures "taste" using matrix factorization, similarly to the method used in recommender systems. Our model enables us to widely explore the parameter space, and derive a set of metrics providers can use to determine whether populating caches according to regional of global tastes provides better cache performance.**

I. Introduction

Internet video today constitutes more than 60 percent of all consumer Internet traffic, with more than 50 percent of this traffic crossing a content-delivery network [16]. Much of this traffic originates from streaming video services such as Netflix, Hulu, HBO GO, Apple TV, and Amazon Prime. In the United States, Netflix now accounts for over 30% of all downstream internet traffic [15]. With the introduction of high density video formats such as 4K, this figure only looks to grow. When a requested movie is retrieved from a remote data center, user experience can degrade due to increased latency and congestion. Video distribution services, such as Netflix, seek to mitigate such performance issues by pushing content closer to their customers using content distribution networks or by deploying their own cache to regional ISPs [13]. These regional caches can store and serve movies to local users, thus avoiding the need to retrieve requested content from an origin server in a distant data center.

Given a large video catalog and multiple encodings of each video [5], only a portion of all movies in the catalog can be stored in a local cache. In this case, a streaming service may intentionally push (or *prefetch*) files into local caches during off-peak hours, based on predicted demand. Netflix, for example, performs a nightly push of the nationally most-popular movies to all its regional caches [13]. This intentional *placement* of video content into local caches differs from demand-driven caches, where requested content that is not found in the cache is downloaded to the cache (typically replacing content in the cache). In the case of intentional placement, a user request not satisfied in the local cache is directed to, and served by, a remote origin server.

But how should the specific content to be placed in local caches be determined? Recent work has shown that geographic locality is correlated with entertainment video consumption [3], [7], and YouTube videos [6]. Intuitively, regional preference may exist because people share common work, activities, social background, language, or social institutions. Such locality in interest suggests that streaming services that utilize local caches could realize substantial benefits (in the form of increased cache hit rate) by determining which content to store and serve in the local cache based on the preferences of local users, rather than the preferences of the global or national customer base.

In this paper, we develop a set of metrics and a methodology that a video service provider (VSP) could use to answer the following questions: How similar are video preferences within a geographic region? How different are such preferences from those of the global population? Is there enough difference across regions to warrant managing cache content based on regional preferences, rather than the simpler solution of loading a local cache with the "globally most popular" videos?

Determining which content to place in a regional cache is complicated for several reasons. Metrics are needed that capture the amount of similarity of user preferences within a region, as well as dis-similarity across regions. A model of user preference must allow for the fact that user preferences in a particular region will likely be (as we will see) a mix of globally-popular content and regionally-popular content. This suggests that hybrid caching policies that consider both globally- and regionally-popular content may be beneficial. A method is also needed to connect these similarity metrics to cache hit performance, our primary performance metric, since high cache hit rates lead to a better user experience – an important consideration for VSPs. Other challenges arise because of the multiplicity of parameters that affect cache hit rates - not only user preferences, but also population sizes, the video catalog size and regional cache capacity.

Because of these challenges above and because there is little publicly available data regarding regional viewing preferences, the degree to which regional content-preferences exist and the extent to which they might be leveraged in VSP operations are both unknown. Thus the task of fully exploring the design

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

space of such regionally-focused caching systems requires a synthetic model with parameters than can be varied. Such a model can then be used to answer ”what if” questions, and inform streaming cache-content management.

Our contributions in this paper are threefold. First, we propose a parsimonious model that quantifies the similarity of user preference (“taste”') within a region, as well as dissimilarities across regions. Many VSPs use an algorithm called Matrix Factorization (MF) or its variants as part of their infrastructure for providing video recommendations [11]. We propose a model of user preference that is compatible with matrix factorization in that our model of user preference is defined in latent factor space (used in MF to describe users and movies). Consequently, VSPs that use MF can readily measure the parameters needed in our model and incorporate our techniques into their existing data pipeline. To the best of our knowledge, this paper is the first to present a methodology for managing cache content based upon user-preference characteristics employed in recommendation systems.

Second, we present a methodology to determine the cache contents for a given customer base. A key contribution is an algorithm that generates synthetic workloads of movie demand based on our user model together with the movie profiles already available to VSPs via their MF-based recommendation systems. A broad range of workloads can be generated by varying the extent of regional tastes in the user profiles. Third, we carry out evaluations using a variety of workloads to illustrate when caching based upon regional tastes outperforms caching based solely upon globally popular content.

In our evaluations, we first show when an *optimal* local policy for placing content in a cache (i.e., content tailored to each local cache) outperforms an *optimal* global policy for the MovieLens dataset [8]. These results illustrate that there is room for improvement over a purely global caching policy. We conduct experiments to study the influence of the many parameters that affect the cache performance. We show that when the regional clusters are far apart (in latent factor space), and the variance of user profiles within a cluster is small, then local caching outperforms global caching. The Movielens data provides one single workload sample, however we show that our model is valid for this sample in that when we compute our key metrics based on the MovieLens data, we can correctly predict whether local or global caching performs better.

Section II defines the problem we address, and Section III details our approach to regional taste modeling and presents an algorithm for generating samples of regional demand. Our experimentation is discussed in Section IV and illustrates which caching policy performs best for a variety of scenarios. Section V discusses related past research, and our conclusions together with directions for future work are outlined in Section VI.

II. Problem Formulation

A. Network Setting

The problem setting for our work is that of a movie streaming service with data centers that hold the complete collection of movie offerings (the catalog), and a user population distributed across a large geographic area. In order to reduce data-center load and provide low-latency user access to content, the streaming service deploys storage devices (“regional caches”) that can hold part of that catalog, typically located geographically close to users.

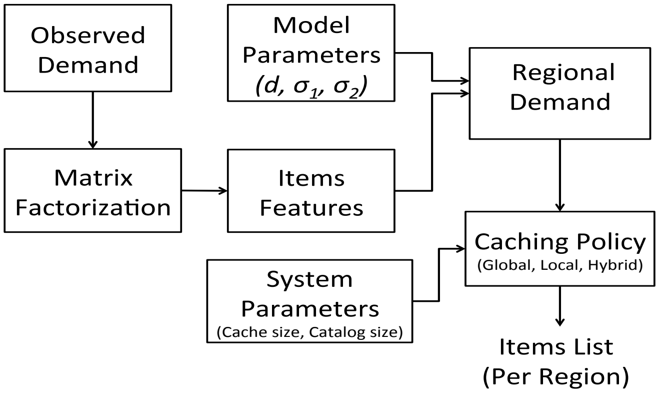


Fig. 1. Overview of our approach

When a user requests a movie, the service first checks whether the movie exists in the regional cache associated with that user. If so, the movie is streamed from that cache; otherwise, the movie is streamed from the central data center. We assume that regional caches are storage-limited, but not processing limited. Since regional caches have limited storage capacity, some requests will “miss” in the regional cache.

Regional caches are *not* demand-driven, i.e., cache misses streamed to the user from the data center are not stored in the regional cache. Instead, regional cache content is determined by the movie streaming service based on movie demand forecast, and uploaded into the cache at specific times, e.g, during the night, when bandwidth use is less expensive [13]. A principal goal of cache content management is to find, for each regional cache, the subset of the movie catalog to store in it, so as to minimize the number of cache misses.

B. Capturing regional tastes

One goal of this paper is to develop a model of user movie tastes that explicitly captures the notion of similarity of user taste within a region (intra-region) and across regions (inter-region). This model can then be used to asses when it is advantageous to determine regional cache content on the basis of that *region's* content preference, and when it is advantageous to set cache content based on the preferences of the *global* user population. By designing a model in which intra-region and inter-region taste similarity are tunable parameters, we can study the effect of similarity in local tastes on cache hit performance. We can also ask ”how much similarity in regional tastes needs to exist for local caching to make sense?”

Matrix Factorization (MF) algorithm (or one of its variants) is a common approach for building recommendation systems [4], [11]. MF is a well accepted model that captures both user and item profiles. Using these profiles, MF fills in missing elements in a user-item ratings matrix (called matrix completion), effectively producing estimates for how a user will rate a movie. Matrix Factorization has shown strong results on both Netflix [10] and MovieLens data [17]. We explain MF, and a common extension called biased matrix factorization (BMF), in Section III-A.

Matrix factorization typically starts with a very high dimensional input matrix with dimensions of number-of-users (in the millions) by number-of-movies (in tens of thousands). This data is converted to a lower dimensional space and the user and movie profiles are defined in that space, usually called *latent factor space*. We thus propose a parsimonious three-tuple model, that can generate user profiles in latent-factor space, where this 3-tuple provides tunable parameters. The three parameters are informally defined as follows.

- **inter-region distance** d**.** The parameter d captures how close or far apart the clusters of regional users are from one another in latent factor space. Intuitively, a small value of d implies that the content preferences of multiple regions are similar - captured by close latent factors - while a larger value of d leads to more distinct regions with separated preferences.
- **inter-region variance** σ_1. The center of each cluster will be selected by generating random values of d from a distribution whose standard deviation is given by σ_1. This parameter thus captures the variance among cluster centers. Small values of σ_1 would indicate overlapping tastes across regions, while large values imply the opposite.
- **intra-region variance** σ_2**.** This parameter is the standard deviation of user preferences in latent-factor space within a region. A small value of σ_2 implies there is much similarity in user taste within that region.

C. Experimental Methodology

The other problem explored in this paper is that of incorporating such a model into an experimental methodology so that a VSP could determine, for their own data, what is the best approach to caching. An overview of our method is given in Figure 1. The method allows a VSP to ask "if my customer base changes and one of these 3 parameters shifts, would I need to change caching strategies?" To do this they need to be able to generate samples of regional movie requests, over multiple experiments, in order to evaluate cache performance. In our method VSPs would characterize their user base by measuring (d, σ_1, σ_2) within their own data. A VSP profiles its movie by extracting item features from an MF type analysis. Note that our solution does not require a VSP's recommendation engine to use MF, as they could simply use MF as a separate standalone tool. However if they already have it implemented, then extracting our three parameters is straight forward. One of our key contributions is an algorithm that takes these inputs (3 user metrics, and movie item features) and outputs sample movie demands. These demands can be used in conjunction with cache and catalog sizes (can also be varied) to run experiments with different caching policies, that compute cache hit rates for the given demand.

Some movies, such as the blockbusters, are likely to be very popular in all or most regions; we refer to these movies as globally popular movies. Other movies may only be well liked within a small number of regions, and we refer to these as locally-popular movies. It is intuitive to expect that user tastes within a geographic region are likely to be a combination of some globally popular movies plus some of local or regional interest. (We will see evidence of this later.)

III. Our Approach

A. Biased Matrix Factorization

The typical input in movie recommendation systems comes in the form of a matrix $\boldsymbol{A}$ that is ($\mathbb{R}^{m\times n}$) where the m rows correspond to users and the n columns correspond to items or movies. Each element contains the rating, on a scale from 1 to 5, that user i gives movie j. The general form of matrix factorization assumes that a matrix $\boldsymbol{A}$ is of low rank and can be decomposed as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}$ where $\boldsymbol{U} \in \mathbb{R}^{m\times k}$, $\boldsymbol{V} \in \mathbb{R}^{k\times n}$, and $k \ll \min(m, n)$. When there are no missing values of $\boldsymbol{A}$, finding $\boldsymbol{U}$ and $\boldsymbol{V}$ can be done by taking the singular value decomposition of $\boldsymbol{A}$. In real world movie and TV recommendation systems, most users have not watched or rated the vast majority of movies (e.g., that can be in the tens of thousands) thus $\boldsymbol{A}$ has missing values, and is in fact typically quite sparse. In this case, $\boldsymbol{U}$ and $\boldsymbol{V}$ can be learned using a method such as conjugate gradient on the known values of $\boldsymbol{A}$. Each row of $\boldsymbol{U}$, namely $\boldsymbol{u_i}$ constitutes a user profile, while each column of $\boldsymbol{V}$ constitutes an item profile. The vectors $\boldsymbol{u_i}$ and $\boldsymbol{v}_j$ are k-dimensional vectors denoting the latent factors of user i and item j respectively. The dot product of these profiles, $\boldsymbol{A}_{i,j} =< \boldsymbol{u}_i, \boldsymbol{v}_j >$, gives a prediction of how user i would rate movie j.

When users' movie tastes are vectors defined this way in latent factor space, the interpretation is that each dimension can be thought of as a meta-genre and the portion of the user latent-factor vector that lies in that space is how much emphasis the user's movie choices place on that genre. For example, the first dimension in the space could correspond to the amount of action in a movie. A user with a high value along this dimension might gravitate towards high-action movies, which would have similarly high values along this dimension, while another user with a low or negative value would actively shy away from them. MF usually creates its own user profiles given the input data $\boldsymbol{A}$. In our scenario we want to create the user profiles separately so we can control and vary the user profiles according to variability in regional taste, namely (d, σ_1,σ_2). As far as we know, we are the first to propose modeling inter/intra-region similarity within the context of matrix factorization, i.e. in latent factor space.

Biased matrix factorization (BMF) assumes that predicted ratings incorporate a number of biases, such as a global bias, user and item biases. This leads to a rating prediction of the form $\boldsymbol{A}_{i,j} = b + \boldsymbol{g}_i + \boldsymbol{h}_j + < \boldsymbol{u}_i, \boldsymbol{v}_j >$. The global bias b is the average offset of the ratings. Because users tend to rate more positive items than negative ones, this is larger than the middle of the ratings scale. The bias $\boldsymbol{g}_i$ assigns a bias to each user based upon whether that user tends to rate movies higher or lower than the average user, while $\boldsymbol{h}_j$ is the item bias and its value indicates whether a movie is generally rated higher or lower than other movies.

B. Generating Synthetic Regional Demand

Our approach for generating movie request data is to (i) create a set of regions, (ii) create a set of users for each region, and (iii) create a set of movie requests for a region based on its users preferences. Our detailed algorithm is defined in Algorithm-1 and we now describe each of the steps.

The first step (1a) in generating a region is to generate its centroid, c_r. Note that in our model, the average user profile corresponds to the origin in latent factor space because we used BMF to explicitly incorporate various biases. By generating movie rating predictions for each user using $\boldsymbol{A}_{i,j} = b + \boldsymbol{g}_i + \boldsymbol{h}_j + < \boldsymbol{u}_i, \boldsymbol{v}_j >$, we have effectively removed biases from the user profiles. Hence the average user profile across all regions sits at the origin in latent factor space. Our parameter d is defined relative to this origin. Each regional centroid is generated by picking a direction vector from a uniform distribution, and then selecting a value for d_r from a Normal distribution with mean d and standard deviation σ_1. This can be interpreted as follows: the profile of an average user in region r (i.e., the regional center) is a distance d_r from the average global user (i.e., the origin in this space). An alternate way to capture inter-cluster distance is to use the pairwise distance between cluster centers. We prefer our definition because of this intuitive interpretation capturing the distance between an average user in a cluster and the globally average user.

A conceptual depiction in two dimensions of these parameters is given in Figure 2. The parameter d corresponds to the average distance between regional centroids and the average user profile. A larger value of d creates regions with more distinct movie tastes. The inter-region variability σ_1 adds noise to the location of regional centroids. Note that when σ_1 is set to 0, then the regional centroids fall on a hypersphere of radius d about the origin. A large σ_1 leads to a set of regions that are a mixture of close to or far from the d-radius hypersphere and increasing this value leads to more diverse regional tastes.

Given the centroid for a region, the next step is to populate it with users. Each user's latent factors are created by drawing samples from a multivariate Gaussian with mean c_r and standard deviation σ_2. The σ_2 parameter can be seen as determining an effective radius for the distribution of users within a region. A large σ_2 leads to increased dissimilarity of user tastes within a region.

Step 2 of our algorithm computes the dot product of user and movie latent factors in order to estimate missing ratings. The final step 3 is to aggregate these per-user per-movie ratings to an indication of regional demand. The regional demand is used both as input to our caching policy and also for experimentation to compute cache performance. In order to run experiments to compute cache hit rates, we need a method of generating movie requests. Since there is no notion of time in our models, or in a typical user-item rating dataset, we create a set of movie requests by doing weighted sampling from popularity ranking of movies within a region.

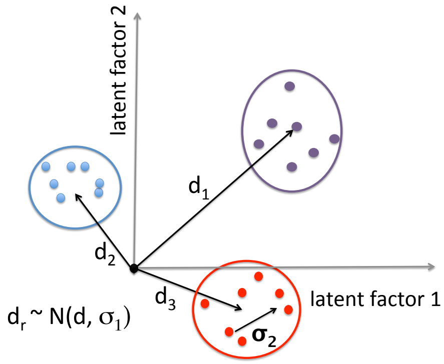
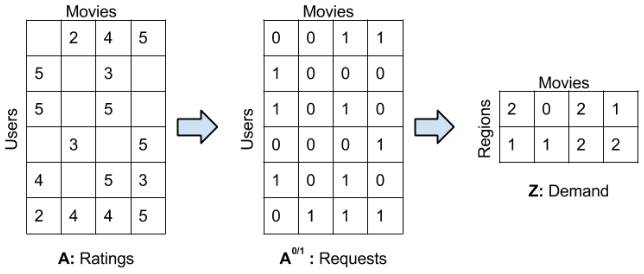


Fig. 2. Conceptual view of user clusters in latent factor space, as a function of parameters (d, σ_1, σ_2).



A: Ratings

	2	4	5
5		3	
5		5	
	3		5
4		5	3
2	4	4	5

A$^{0/1}$: Requests

0	0	1	1
1	0	0	0
1	0	1	0
0	0	0	1
1	0	1	0
0	1	1	1

Z: Demand

2	0	2	1
1	1	2	2

Fig. 3. An example ratings matrix $\boldsymbol{A}$ with missing values, a corresponding requests matrix $\boldsymbol{A}^{0/1}$, and a condensed regional demand $\boldsymbol{Z}$ with each of the regions containing 3 users (users 1-3 and 4-6 respectively).

A ratings matrix is converted to regional demand as follows. This user-item rating matrix $\boldsymbol{A}$ is converted into a binary matrix ($\boldsymbol{A}^{0/1}$) where $\boldsymbol{A}^{0/1}_{i,j} = 1$ corresponds to user i requesting movie j while a $\boldsymbol{A}^{0/1}_{i,j} = 0$ means the movie has not been requested. Movies can be marked as requested if their rating prediction exceeds a threshold μ. The idea here is that only movies that are highly rated are likely to be requested again or by new viewers. To create a variety of samples of demand matrices a movie is only included in the sample with a probability p_j that captures its general popularity. We then use $\boldsymbol{A}^{0/1}$ to count the number of requests within a region for each movie. This is done by summing the rows of users belonging to the same region so that $Z_{r,j}$ corresponds to the number of users in region r who have watched movie j. The rows of matrix $\boldsymbol{Z}$ represent local movie demands while the sum down the columns of the matrix correspond to the global demands of each movie. This transformation is depicted in Figure 3. Our notations are summarized in Table III-C.

Algorithm 1 Generation of Synthetic Regional Demands

```
Given: $n$, $R$, $d_r$, $\sigma_1$, $\sigma_2$, $\boldsymbol{h}$, $\boldsymbol{V}$, $\boldsymbol{p}$, $\mu$
  $\boldsymbol{p}$     Vector of global movie popularities
  $\mu$     A threshold to determine whether a user likes a
            movie enough to request it

  1) For 1 to R:
      a) Generate Regions: Pick a random direction uniformly. Sample a distance from a normal distribution with mean $d$, and standard deviation $\sigma_1$. Treat this new point $\boldsymbol{c}_r$ as the centroid for region $r$.
      b) Generate Users: For 1 to n:
          i) Sample the latent factors $\boldsymbol{u}^*$of a user belonging to this region from a multivariate normal distribution with mean $\boldsymbol{c}_i$ and standard deviation $\sigma_2$
  2) Predict Movie Ratings: Let $\boldsymbol{A}^*_{ij} = \boldsymbol{h}_j + < \boldsymbol{u}^*_i, \boldsymbol{v}_j >$
  3) Convert Predictions to Regional Requests
     Let $\boldsymbol{A}^{0/1}_{ij} = \begin{cases} 1 & \text{with probability } \boldsymbol{p}_j \text{ if } \boldsymbol{A}^*_{ij} > \mu \\ 0 & \text{otherwise} \end{cases}$
  4) Let $\boldsymbol{Z}_r = \sum_{\text{user i in region r}} \boldsymbol{A}^{0/1}_i$
  5) Return $\boldsymbol{Z}$
```

Notation	Definition
$\boldsymbol{A}$	$\mathbb{R}^{m\times n}$ user-item ratings
$\boldsymbol{A}^{0/1}$	$\mathbb{Z}_2^{m\times n}$ user-item requests
$\boldsymbol{Z}$	$\mathbb{Z}^{r\times n}$ region-item demand
λ	$\lambda \in [0,1]$ balance between local and global preferences
b	The global bias
n	The population per region
$\boldsymbol{g}$	$\mathbb{R}^m$ user biases
$\boldsymbol{h}$	$\mathbb{R}^n$ item biases
$\boldsymbol{U}$	$\mathbb{R}^{m\times k}$ user latent factors
$\boldsymbol{V}$	$\mathbb{R}^{k\times n}$ item latent factors
d	The mean distance from regional centroids to the global mean use profile.
σ_1	Inter-region variability: standard deviation of cluster centroids
σ_2	Intra-region variability: standard deviation of user profiles within a region
d_r	Generated from a $\mathcal{N}(d, \sigma_1)$ distribution
$\boldsymbol{c}_r$	Centroid of region r
$\mathcal{C}$	Size of the regional cache (number of movies stored)

TABLE I
NOTATION USED THROUGHOUT THE PAPER.

C. Bootstrapping with Empirical Inputs

Our algorithm needs certain inputs that characterize the movies, in particular it needs the item biases $\boldsymbol{h}$, the movie profiles in latent factor space $\boldsymbol{V}$, and some global popularity statistics $\boldsymbol{p}$. To get real world estimates for these values, we use the public MovieLens dataset. In particular, we use the data that contains roughly 6000 users, 4000 movies, and 1 million user-movie ratings between 1 and 5. The item biases $\boldsymbol{h}$ and item factors $\boldsymbol{V}$ are calculated by running BMF once on this dataset, and they do not change.

Although running BMF will generate user profiles as well, we do not use them. We point out that any service provider who uses a snapshot of their own data, will produce one set of user profiles - and hence only one specific triple (d,σ_1,σ_2). Hence a VSP cannot use this data to answer questions like ”would I have to change my caching policy if σ_2 shrunk? or if d increased a little?” A synthetic model can do just this, and hence our algorithm can consider changes to the triple (d,σ_1,σ_2), and generate new user profiles to reflect these changes, and then generate samples of what the regional demand would be under a change.

The lack of public data available to get different snapshots (e.g. from different providers) makes it hard to understand the extent or variety of (d,σ_1,σ_2) across different services. The single sample that the MovieLens data provides us is as follows: ($d = 0.2$, $\sigma_1 = 0.07$, $\sigma_2 = 0.34$). We computed this by grouping users into 60 clusters. (Note we use the terms clusters and regions interchangeably.) We selected 60 because this data is from the US and this roughly captures the 50 states; since a few states are very large with multiple metropolitan areas (e.g. Florida, Texas, California, etc.), we elected to use a few more clusters than 50. On the one hand, we wanted to cover the entire US region, but we also wanted to limit some geographic areas to not be too big. By using 60 we ended up with roughly $n = 100$ users per region. (Note that we were unable to use the 10M Movielens dataset because that data does not contain zip codes and thus we could not geolocate the user base.)

D. Hybrid Caching Policy

We consider a hybrid caching policy, as specified in Algorithm 2. This algorithm takes the regional demand computed in Algorithm 1. Using this demand, each movie, in each region, is assigned a score based on a linear combination of the movie's global popularity and local popularity. Each cache is then populated with the top $\mathcal{C}$ movies that have the highest weighted popularity. The score is influenced by a parameter λ that varies from 0 to 1. When $\lambda = 0$ the policy corresponds to a cache filling policy based solely on global tastes, and this creates identical caches in each region. $\lambda = 1$ corresponds to a policy based solely on regional tastes, and values of λ in between fill the cache with some of each type. The case of $\lambda = 0$ captures the practice adopted by some in the industry [13]. For the sake of simplicity, we experiment with scenarios in which we assume that all movies are the same size and each uses one unit of capacity of the cache size $\mathcal{C}$. Note that Algorithm 2 defines our method for selecting which items from the regional demand go into the cache. When we use it in experiments, we use 80% of the demand to fill the cache, and the remaining 20% are used for testing.

IV. RESULTS

In this section, we analyze the effect of our model parameters (d, σ_1, σ_2) as well as system parameters (cache size $\mathcal{C}$, the size of the population per region n) on the preferred caching policy. We begin with computing the specific model parameters of a real-world dataset, MovieLens, and analyzing the

Algorithm 2 Hybrid Caching Policy

```
Given: $\boldsymbol{Z}$, $\lambda$, $\mathcal{C}$

1) Calculate $\boldsymbol{Z}^*$: assign score to each movie for each region
       $\boldsymbol{Z}^*_{rj} = \lambda \boldsymbol{Z}_{rj} + \frac{(1-\lambda)}{R} \sum_{k=1}^{R} \boldsymbol{Z}_{kj}$
2) For each region $r$, rank the movies in descending order by $Z^*_{rj}$
3) Select the top $\mathcal{C}$ ranked movies to fill the region's cache
```

performance of different caching policies using this dataset. We then evaluate different model parameters and study how each affects the cache hit rate and discuss which caching policy operates best in each model setting.

Computing Model Parameters from MovieLens. The MovieLens dataset provides us with an approximation to movie demand and the ability to derive movie features. In particular, we use the MovieLens 1M dataset that contains roughly 6000 users, 4000 movies, and 1 million user-movie ratings, each between 1 and 5. We computed the latent user features $\boldsymbol{U}$, movie features $\boldsymbol{V}$ and movie biases $\boldsymbol{h}$ by running BMF on this dataset.[1] We selected the dimension of the latent factors to be 10, with 20 iterations, learning rate of 0.15 with 0.95 decay per iterations. These values were empirically shown to perform well for this dataset [4].

The MovieLens dataset contains zipcode for each user, which enables us to assign users into regions. We converted each zipcode into lat-long coordinates using Boutell's Zipcode dataset[2], and clustered the users into geographic regions using spectral clustering. We empirically set the number of clusters to 60, each containing roughly 100 users. We selected 60 since it places users in all major states, and also breaks down the large states, e.g., Florida, Texas, California, into smaller, more uniformly populated regions. Once we assigned users to regions, we used the latent user features to compute the model parameters, and found $d = 0.2$, $\sigma_1 = 0.07$, $\sigma_2 = 0.34$. We note that d and σ_1 are quite small, indicating that the users in MovieLens across different regions have rather similar taste in movies. To verify this, we looked at the demographic distribution of the MovieLens population, and found that 71% of the users in the dataset are males and 72% are young adults (18–35 years old). Furthermore, the most common stated occupations of the users are executives and engineers (40%), followed by college students and academics (10%), and less than 5% of the users state that they have low-education jobs. These biased distributions explains the model parameters, resulting in small d and σ_1 and a large σ_2. Furthermore, it encourages us to explore different values of the model parameters that cover more diverse populations.

Next, we evaluate the performance of the different caching policies. To this end, we split the dataset into an 80% training set (observed demand) and 20% evaluation set (future demand). We consider two optimal policies, global and local, that have perfect knowledge of future requests. The *Optimal Local Policy* fills each regional cache with the movies desired by the users of that region. The *Optimal Global Policy* fills all caches the same way with the top popular movies across the entire customer base. In addition, we consider our hybrid policy with varying values for λ. In all policies, the movies are order by decreasing demand and the caches are filled up to capacity, denoted by C.

Once the caches are filled based on a policy, we simulate the demand using the 20% evaluation set, and measure the hit rate, i.e., fraction of cache hits out of all requests for each cache, and averaged across all 60 caches. We repeat this experiment 5 times, each using a different split of the dataset to training and evaluation sets. Figure 4 shows the cache hit rate averaged over the 5 experiments. The plot shows, as expected, that the hit rate increases with cache size. The optimal local policy provides the best hit rate results, consistently outperforming the optimal global policy. This indicates that it is possible to outperform global only policies by using local tastes as long as regional preference information can be captured in a useful way.

The (non-optimal) "Global Policy" ($\lambda = 0$) performs very similarly to the optimal global policy, because the dataset as a whole has enough samples to estimate the globally popular movies reasonably well. Furthermore, our simulated future demand is actually drawn from the same time-frame as the observed demand, thus it is relatively simply to "predict" the future. This leads the cached movie list obtained from the optimal global policy to be almost identical to the one obtained from the non-optimal policy.

However, our hybrid policies do not perform as well, especially for high locality (e.g., $\lambda \geq 0.75$); not only are these far from the optimal local policy but they perform worse than the pure-global policy. To better understand this result we manually inspected the top 2000 locally-popular movies in several regions. We found that roughly the top 500 movies correspond to ones in the globally popular movies set. This limits the potential benefits of using local rather than global lists. The second problem is that the partitioning of the dataset into regions provides limited samples when computing the locally popular movies. Overall the MovieLens model parameters exhibit a skew towards a more global "taste", hence a global caching policy performs well.

Modeling Demand. Next, we explore the model parameter space to understand when local preference begins to indicate a different caching policy should be used (i.e. locality-aware rather than a global only policy). To study the hit rate obtained from a certain policy, we need to simulate a population that follows the model parameters, and generate both observed and future demand. For a given experiment, we first use the model parameters (d, σ_1, σ_2) to generate regional user profiles. We then use the MovieLens movie features together with the

[1] Although we experiment with BMF on this medium-sized dataset, BMF is known to scale well in terms of users and movies for large datasets which is why it has been adopted in practice.

[2] http://www.boutell.com/zipcodes/

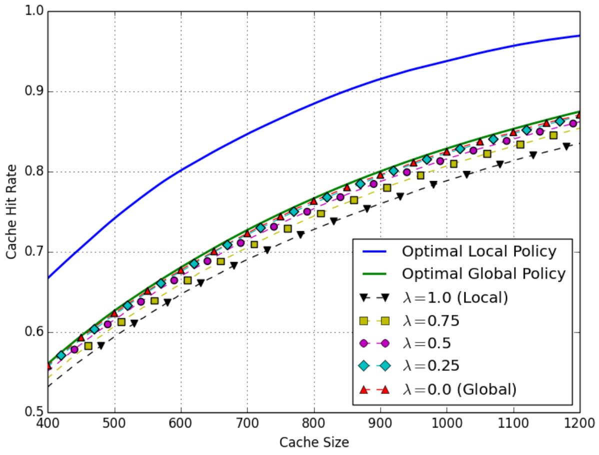


Fig. 4. Cache hit rate using various caching policies on MovieLens data. Each movie uses 1 unit of cache capacity. MovieLens data constitutes single sample of inter- and intra-region tastes.

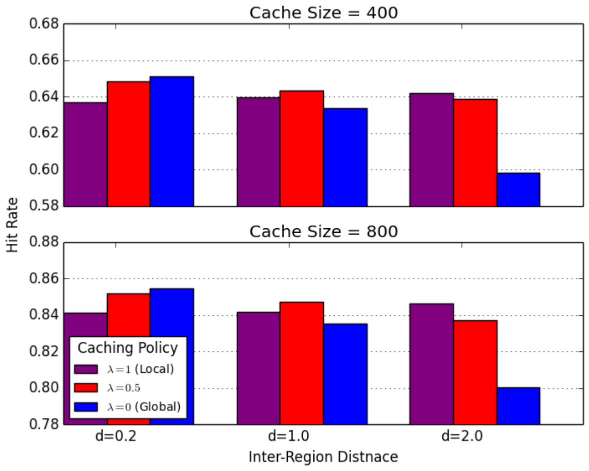


Fig. 6. Comparison of three caching policies for various inter-region distances. Model parameters: $n = 100$, $\sigma_1 = 0.07$, $\sigma_2 = 0.34$

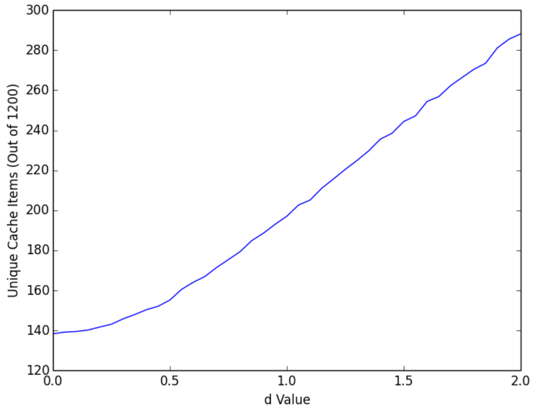


Fig. 7. The set difference between a cache filled based on a local policy (λ=1) and one filled by a global policy ($\lambda = 0$) as d increases. Cache size is 1200 movies.

MovieLens movie popularity distribution to create a sample rating matrix A. We then convert A into a demand matrix $A^{0/1}$ as follows. We convert ratings predicted to be 4 or 5 as '1', and the rest become '0' (this corresponds to setting $\mu = 4$ in Algorithm1). This is simply because in most 5-star rating systems, a 4 or higher indicates the user liked the movie, and thus we consider others with similar taste may request it. We could have used 3 as a threshold and we experimented with 3 as well. The overall results were the same, and thus we only include the results for a threshold of 4 herein, due to lack of space. We mark 80% of the demand as the observed requests and the remaining 20% as the future requests, which are used to compute the cache hit rate.

Inter-Region Distance. Recall that our model parameter d captures the distance between the global mean of all users profiles and the center of the regions. Intuitively, a small d, e.g., such as the one in MovieLens, will results in regions that share similar taste with the global "mean user". This implies that local caching policies will not be beneficial. Conversely, a large d will cause regions to be far apart, and hence will favor local caching policies over global.

Figure 5 shows the hit rate for increasing cache size for different values of d. As noted in Section III-D, each movie consumes one unit of the cache capacity. In these plots, σ_1 and σ_2 are held constant at values of 0.07 and 0.34. For small values of d the models perform comparably, and as we increase d the local $\lambda = 1$ and hybrid policies outperform the global policy $\lambda = 0$. Figure 6 further illustrates this for two cache sizes. When d is small the global policy performs best, when $d = 1.0$ the regions begin to exhibit different regional "taste", but there is still some overlap, making the hybrid policy best. When d becomes sufficiently large, each region has unique tastes which causes the purely local caching policy to outperform the rest.

Finally, to quantify how different the cached movies are, resulting from a local or global policy, we compute the set difference between the two caches, i.e., the number of movies that are unique to a each region. Figure 7 plots the average set difference across all regional caches, averaged over 10 experiments. The figure shows that as d increases, the number of unique movies per regional cache grows, indicating that the regional "taste" is increasingly different than the global average.

Population of a Region. Intuitively, highly populated regions with distinct taste benefit most from local caching policy. To study the joint effect of inter-region distance and population size of a region, Figure 8 plots the percent improvement between purely local and purely global caching policies for different population sizes n and values of d. Negative values indicate that global caching outperforms local. The white dot corresponds to the MovieLens parameters, exhibiting that local caching yields a 2% lower hit rate that a global policy.

The figure shows that for low values of d, increased population does not change the caching policy decision. However,

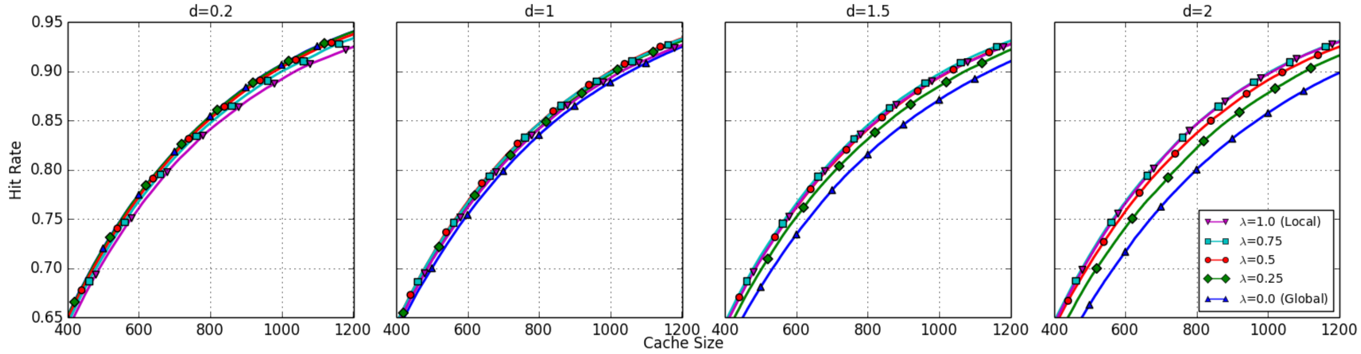


Fig. 5. The effect of inter-region distance d on cache hit rate. Model Parameters: $n = 100$, $\sigma_1 = 0.07$, $\sigma_2 = 0.34$

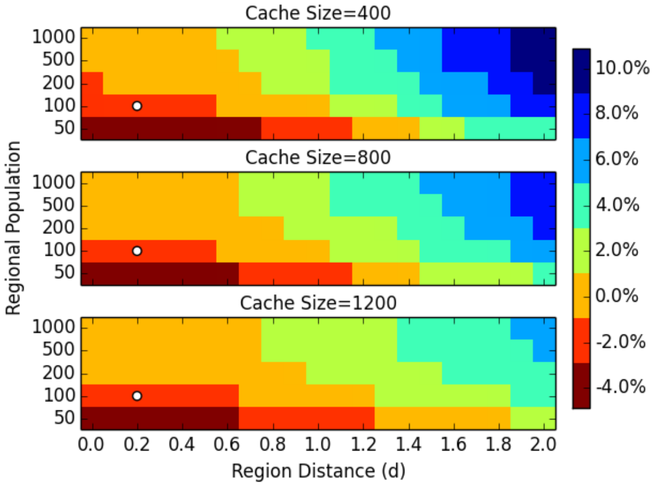


Fig. 8. Color shows cache hit rate gain of local over global caching. The white dot shows the value of our model parameters for the MovieLens data. Model variability parameters are $\sigma_1 = 0.07$, $\sigma_2 = 0.34$. Cache sizes of 400, 800 and 1200, correspond to 12%, 24% and 36% of the catalog size.

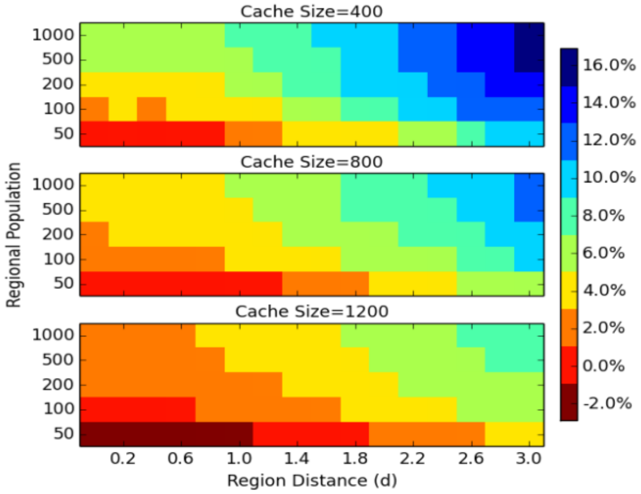


Fig. 9. Each color shows the cache hit rate gain using local caching instead of global. Model variability parameters are $\sigma_1 = \sigma_2 = 0.5$. Cache sizes of 400, 800 and 1200, correspond to 12%, 24% and 36% of the catalog size.

as d increases, local caching policy is preferred even for lower population size, reaching 10% higher hit rate than the global policy. As expected, for large d, highly populated regions clearly prefer local caching policies, as the number of requests that can be served by a locality-aware caching policy increases. The figure also shows that as the cache size increases, the gain of local caching diminishes, although still remains positive for roughly half of the parameter space depicted here. As cache size increases (in this case reaches 36% of catalog size), the cache is able to hold all movies, both of global and local taste that a region needs, and so the difference between the two schemes is somewhat diminished.

Inter-Region and Intra-Region Variability Recall that in our model σ_1 also contributes to inter-region variability along with d, and σ_2 captures the intra-region variability within each region. In Figure 8, we varied d and n while holding $\sigma_1 = 0.07$ and $\sigma_2 = 0.34$ constant; we used these MovieLens parameters to illustrate where the MovieLens data falls in the parameter space. Figure 9 shows a heatmap for other values of the inter- and intra-region variabilities, namely $\sigma_1 = \sigma_2 = 0.5$. The importance of these two parameters is highlighted here as the results are quite different - with local caching outperforming global caching for lower d than in the previous plot.

Figure 10 plots the percent difference between purely local and purely global caching policies for varying values of σ_1 and σ_2. As the plot shows, when σ_1 is high (large variability between regions) or when σ_2 is low (distinct "taste" of each cluster), the locality-aware caching policy significantly outperforms the global policy, reaching up to 12% improvement in hit rate. Conversely, when σ_2 is high (users within a region differ a lot), and σ_1 is low (clusters have overlapping preferences), then global caching is the best solution.

V. Related Work

There have been numerous studies of demand-driven cache-content management in web and VoD systems where content streams through a cache, which must then dynamically decide which content to maintain in the cache [14], [12], and specifically in a commercial VSP setting [2]. Here, we consider non-

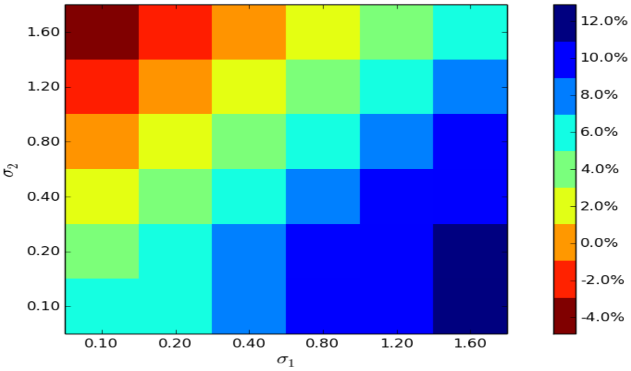


Fig. 10. The Effect of Inter-Region and Intra-Region Variance. The color of the plot shows the hit rate gain using Local Caching over Global Caching. Model Parameters: $n = 200$, $d = 1.0$, $\mathcal{C} = 800$

demand-driven caches, focusing on the differences between global versus regional preferences, modeling these "taste" differences with a matrix factorization framework. The system operation of one such VSP was studied in [1]. Considering regional preferences as a component of a CDN federation analysis was performed in [3].

The editorial note of [9] is perhaps the closest in spirit to our research – at the boundary of recommendation systems and content management in networked cached VSP systems. They suggest the potential use of recommender system techniques to predict demand and determine how to replicate content among caches, but do not investigate the use of specific recommendation system techniques nor perform a quantitative investigation. Here, we specifically use MF to incorporate regional preferences in a parsimonious model for aiding VSPs in designing cache-content management techniques; we also quantitatively demonstrate the value of use of this model using a "real-world" dataset.

VI. Conclusions and Future Work

Streaming video dominates Internet traffic, and service providers must actively push and cache content closer to the users to provide better customer experience and lower load on their infrastructure. In this paper we study the problem of deciding which content to cache, and the impact that regional movie preferences have on this decision. We propose three important metrics for VSPs to measure, on their own customer profiles, to quantify the extent to which regional tastes are present. We provide a modeling framework, that uses these metrics, along with an algorithm for generating synthetic workloads of regional demand, to assess whether or not caching policies should leverage regional tastes.

Overall, we show that when inter-region distance and variability are large, or intra-region variability is small, our model produces well-defined regional preference that vary across regions - and our experimentation framework clearly demonstrates the benefits local caching policy over a purely global caching strategy. As the inter-region distance and variability grow, the benefits of local caching increase. Moreover, we saw that increased population sizes per region are helpful is capturing regional taste, and can affect the choice of caching policy. In the future, we plan to study the influence of clustering and consider other performance metrics such as bandwidth costs. We plan to further extend our model to take into account changes in movies popularity over time. This will enable us to better forecast future demand of movies, and potentially design better caching policies.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-1117764. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] V.K. Adhikari, Yang Guo, Fang Hao, M. Varvello, V. Hilt, M. Steiner, and Zhi-Li Zhang. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628, March 2012.

[2] David Applegate, Aaron Archer, Vijay Gopalakrishnan, Seungjoon Lee, and K. K. Ramakrishnan. Optimal content placement for a large-scale vod system. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 4:1–4:12, New York, NY, USA, 2010. ACM.

[3] Athula Balachandran, Vyas Sekar, Aditya Akella, and Srinivasan Seshan. Analyzing the potential benefits of cdn augmentation strategies for internet video workloads. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 43–56, 2013.

[4] Smriti Bhagat, Udi Weinsberg, Stratis Ioannidis, and Nina Taft. Recommending with an agenda: Active learning of private attributes using matrix factorization. In *RecSys*. ACM, 2014.

[5] The Netflix Tech Blog. High quality video encoding at scale. http://techblog.netflix.com/2015/12/high-quality-video-encoding-at-scale.html, 2015. Accessed: 2015-12-09.

[6] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. Youtube around the world: geographic popularity of videos. In *Proceedings of the 21st international conference on World Wide Web*, pages 241–250. ACM, 2012.

[7] Wenya Dou, Guanping Wang, and Nan Zhou. Generational and regional differences in media consumption patterns of chinese generation x consumers. *Journal of Advertising*, 35(2):101–110, 2006.

[8] GroupLens. Movielens 1m dataset, 2003.

[9] Mohamed Ali Kaafar, Shlomo Berkovsky, and Benoit Donnet. On the potential of recommendation technologies for efficient content delivery networks. *ACM SIGCOMM Computer Communication Review*, 43(3):74–77, 2013.

[10] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81, 2009.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[12] Yong Liu, Yang Guo, and Chao Liang. A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, 1(1):18–28, 2008.

[13] Netflix. *Netflix OpenConnect Appliance Deployment Guide*, April 2015. Version 3.7. Version 3.7.

[14] Stefan Podlipnig and Laszlo Böszörmenyi. A survey of web cache replacement strategies. *ACM Comput. Surv.*, 35(4):374–398, December 2003.

[15] Sandvine. Global internet phenomena report. https://www.sandvine.com/trends/global-internet-phenomena/, 2011.

[16] Cisco Systems. Cisco visual networking index: Forecast and methodology, 20142019. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html. Accessed: 2015-07-27.

[17] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, 2007.