Leveraging Behavioral Heterogeneity Across Markets for Cross-Market Training of Recommender Systems

Kevin Roitero*
University of Udine
roitero.kevin@spes.uniud.it

Ben Carterette
Spotify
benjaminc@spotify.com

Rishabh Mehrotra
Spotify
rishabhm@spotify.com

Mounia Lalmas
Spotify
mounia@acm.org

ABSTRACT

Modern recommender systems are optimised to deliver personalised recommendations to millions of users spread across different geographic regions exhibiting various forms of heterogeneity, including behavioural-, content- and trend specific heterogeneity. System designers often face the challenge of deploying either a single global model across all markets, or developing custom models for different markets. In this work, we focus on the specific case of music recommendation across 21 different markets, and consider the trade-off between developing global model versus market specific models. We begin by investigating behavioural differences across users of different markets, and motivate the need for considering market as an important factor when training models. We propose five different training styles, covering the entire spectrum of models: from a single global model to individual market specific models, and in the process, propose ways to identify and leverage users abroad, and data from similar markets. Based on a large scale experimentation with data for 100M users across 21 different markets, we present insights which highlight that markets play a key role, and describe models that leverage market specific data in serving personalised recommendations.

ACM Reference Format:
Kevin Roitero, Ben Carterette, Rishabh Mehrotra, and Mounia Lalmas. 2020. Leveraging Behavioral Heterogeneity Across Markets for Cross-Market Training of Recommender Systems. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3366424.3384362

1 INTRODUCTION

Recommender systems now touch nearly everything people do on the internet, from entertainment to search to fashion. This means that users of recommender systems are an incredibly diverse group: they range from the very young "digital natives" to people for whom the internet did not even exist until late in their life, and of course from all the different countries in the world and all the different cultural values and expectations that come from that.

With such a large and heterogeneous group, it is unlikely that a single recommendation model will suffice for all users. But in practice, machine-learned models must be trained on *some* set of data, and which set of data is chosen may have a large impact on which users have the best experience with the deployed system. Even if different models are deployed to different segments of the user population, choices must be made regarding how granular to segment users, what data to use for training each segment, how frequently to retrain, etc. Furthermore, some ways to segment may raise the need to have ways to "cold start" new segments—for example, if training is done by market, expanding into new markets will require having ways to train new models for those markets.

Segmenting by market is the main focus of this work. We consider the case of a music recommender system, with over 200 million users in 79 different markets, specifically recommending playlists for user consumption. We investigate the cost-benefit trade-off that is achievable when choosing between global and general models versus multiple localized and custom market-specific models. Our research questions are as follows:

- RQ1 Is it true that market is one of the most important attributes to consider when deploying machine-learned models?
- RQ2 If market is an important attribute, how can we best trade off between training global market-independent models versus training models locally for each market?
- RQ3 What are the best strategies to train a recommender system for a new market?

We show that market is in fact one of the most important considerations when segmenting users for the purpose of training and deploying models, using statistical data analysis of user interaction and system effectiveness. We then detail and consider many different training strategies that allow us to cover the entire spectrum of possibilities: from a single global model to multiple market-specific ones, passing through hybrid possibilities.

2 RELATED WORK

Considering user heterogeneity: The effect of heterogeneity of both users and in particular markets has been studied in the recommender system domain. Works reported in [1, 2, 6, 18, 30] discuss context-aware recommender systems and how to incorporate and model contextual information in the design of a recommender system. In [8], the integration of user behaviour is considered, crawled from a set of Twitter data, in the context of music recommendation. Using a collaborative filtering setting they show that the integration

*This work was done while the author was an intern at Spotify.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3384362

of user diversity features improves the quality of recommendations. In [26], a set of user features is included in the training of a recommender system, which include, among others, diversity and novelty. Based on the evaluation on 200M listening events crawled from a Last.fm dataset, the authors claim that such features improve recommendation effectiveness.

In [16], music diversity across countries is explored using a Last.fm crawl. The results suggest that by using metrics that are country dependent, the overall subjective user evaluation can be maximised. In [18], the effect of the user search mode and the product type is considered in the effectiveness of a collaborative filtering model, with the conclusion that the usage of user features improves the model accuracy.

In social sciences, researchers have long used Analysis of Variance (ANOVA) and related statistical models to understand the effects of heterogeneity on outcomes—the literature is far too wide to cite here, but we refer readers to [17] for examples and advanced models. There has recently been some work applying such methods to Information Retrieval (IR) system outcomes, as we describe next.

Breaking down system components: Understanding how a complex system works has been studied by the IR community. Of particular interest is the evaluation of retrieval systems that retrieve documents from a test collection. This is a challenging problem, usually addressed using large evaluation campaigns. In the test collection setting, multiple (variants of) systems, called runs, retrieve documents from large collections of documents, addressing different (artificial) information needs, called topics. After the retrieval phase, trained human judges assign to a subset of retrieved documents, called pool, a relevance score. The set of (topic, document) relevance scores is then used to compute the effectiveness metrics for each (topic, run) pair. The runs are then ranked using an aggregated score of their effectiveness values over the individual topics. Because we are able to evaluate each topic in the collection against each system variant, we can obtain large amounts of measurement data to use in ANOVA models for understanding the effect of different system components.

Many works have attempted to model and understand the vast majority of possible system configurations. A common way to break down system effectiveness is to use a General Linear Mixture Model (GLMM) [21, 23, 28] together with the Analysis Of Variance (ANOVA) [22, 25], where the overall run performance is described as combination of topic and run component effect, plus their interactions [3, 5, 19, 24, 27, 29]. Different work considered different statistical analysis and techniques, but always breaking down the run and topic effect. More recently, in [13] and [12], GLMM and ANOVA have been used to formalise the sub-corpora effect. GLMM and ANOVA were applied in [14] and [15] to the composition of the run components and their effect, using a Grid of Points (GoP) setting, borrowed from [11], to study all the possible run configurations. Finally, GLMM and ANOVA were used to study the effect of different test collection components on topic difficulty estimation in [28]. To the best of our knowledge, our paper is the first work using GLMM and ANOVA to break down user heterogeneity in the recommender system scenario.

Domain adaptation: Different works tackled the problem of domain adaptation in recommendation systems via transfer learning [4, 10], that is the research area which studies how to transfer knowledge between different domains when delivering recommendations. Semantic networks and knowledge based descriptors have been used to deliver cross-domain item recommendations in [9], whereas a framework to match entities in source and target domain for recommender systems was proposed in [31]. In [7], features sets based on user behaviour were considered in a deep learning model to build a latent space that is optimal for cross-domain user modelling. Finally, the consistency of recommendations across domains is investigated in a collaborative filtering scenario in [20]. We add to this body of work by studying the training and testing of a music recommendation system across 21 different markets.

3 USER-LEVEL HETEROGENEITY ACROSS MARKETS

To effectively recommend songs and playlists in a personalized way, user-related features play a major role in optimizing user satisfaction. In general, it is known that how users interact with an online system can differ depending on demographics, including age, gender, location, etc. [1, 8, 16, 26]. We therefore study user variance and heterogeneity to better understand the components that affect user interaction metrics. Following from this, we deepen our analysis looking at the effect of user heterogeneity in machine learning effectiveness.

3.1 Modeling heterogeneity in user interaction signals

Our aim is to understand the effects of component user attributes that affect user interaction signals; our hypothesis is that the user *market* is one of the most important. To do this, we set up a model of a user interaction signal as a function of these attributes, then fit the model to interaction log data collected over 100M users.

The user attributes, or *factors* we consider are the following:

- *Platform age*: the number of days the user has been registered, divided into buckets;
- *Product*: the kind of product the user is using: free, premium, or other variations;
- *Activity*: the user activity level, discretized into not active, medium active, very active;
- *User age*: the age of the user, divided into buckets;
- *Gender*: male, female, or other;
- *Market*: geographic region of the user.

The set of interaction signals we consider is detailed in Table 1. Each signal is aggregated (by arithmetic mean) over all of the users that are described by a unique ⟨ Age, Product, Activity, User-age, Gender, Market ⟩ tuple.

We combine these factors in a linear model, which allows us to use estimation and inference techniques associated with ANOVA (Analysis of Variance). We write it as follows:

$$\begin{aligned} \text{Interaction}_{ijklmn} = \ & \text{Market}_i + \text{Age}_j + \text{Product}_k + \\ & + \text{Activity}_l + \text{User-age}_m + \\ & + \text{Gender}_n + \text{Error}_{ijklmn} \end{aligned} \tag{1}$$

Table 1: Interaction signals and user factors. For purposes of analysis when fitting an ANOVA model, we use the arithmetic mean of the signal over all users in a group characterized by the tuple of model feature values.

Interaction Signal	Extended Description
interaction time	number of time units (e.g., seconds) a user interacted with the graphic user interface of the application.
stream time	cumulative time units (e.g., seconds) spent by the user streaming platform content within a session.
dwell time	time units spent by the user being active in the user interface but not interacting with it; i.e., time in which no action is performed.
max depth	maximum depth reached in the expanded tree of the application map, correspondent of a sequence of user actions.
ms played	time unit, in milliseconds, spent looking at / streaming platform content within a session.
number of interactions	number of interactions done by the user in the graphical user interface of the application. This metrics includes clicks, scrolls, etc.
shelf interaction	number of interactions with a particular platform object called shelf, which is central for the recommender system.
session length	total time of the length of the user session in the platform application.
items played	number of items the user has streamed.
time to last exit	units of time passed between the current and the previous session of the user.

Table 2: ANOVA table when modeling mean number of interactions. Independent variables are ranked in decreasing order of their estimated effect size ω^2.

factor	SS	df	F	p-val	η^2	ω^2
market	29.08	20	20.60	2.3e-74	1.37e-2	0.0131
platform-age	19.77	4	70.06	3.8e-59	9.33e-3	0.0091
user-age	18.76	5	53.18	3.8e-55	8.85e-3	0.0086
product	13.47	9	21.21	3.4e-36	6.35e-3	0.0060
gender	1.2	2	8.74	1.60e-4	5.82e-3	0.0005
activity	0.02	1	0.40	0.525	1.33e-5	0.0000

Using ANOVA with this model returns quantities such as the mean sum of squares SS, degrees of freedom df, the F-statistic, a p-value for testing statistical hypotheses, η^2 for single-factor model fit (i.e. a measure of model fit like R^2 but for only one factor under consideration), and measures of *effect size* such as ω^2. Effect size is particularly interesting, as unlike model fit or p-values, it is independent of sample size, and thus provides an understanding of the population-wide relationship between a factor and outcomes. The ω^2 effect size is standard for ANOVA and defined on a per-factor basis as follows [15]:

$$\omega^2_{\text{factor}} = \frac{df_{\text{factor}}\left(F_{\text{factor}} - 1\right)}{df_{\text{factor}}\left(F_{\text{factor}} - 1\right) + N}$$

We fit 10 ANOVAs, one for each of the interaction signals in Table 1.

Table 2 shows results of the ANOVA for mean number of interactions, with factors in decreasing order of effect size ω^2. The market variable explains more about the variation in mean number of interactions than any other factor in the model. These results hold over all interaction signals. The market a user is in explains more about their interactions than factors such as age, gender, activity level, and even which product they choose to use.[1]

[1]Note that though we did fit multiple models and therefore would be advised to perform multiple comparisons correction, no such correction would change the relative impact of the factors in any given model.

Table 3: ANOVA table when modeling accuracy. Independent variables are ranked in decreasing order of their estimated effect size ω^2.

	SS	df	F	p-val	η^2	ω^2
feature	7.86	3	1067.35	1.8e-291	0.4291	0.4286
algorithm	5.75	2	1171.94	5.7e-248	0.3141	0.3138
ft:alg	1.09	6	73.73	2.1e-74	0.0593	0.0585
market	0.72	20	14.66	9.4e-43	0.0393	0.0366
mkt:ft	0.60	60	4.04	2.79e-20	0.0325	0.0245
mkt:alg	0.14	40	1.38	0.006	0.0074	0.0020
size	0.02	3	3.26	0.021	0.0013	0.0001

3.2 Modeling heterogeneity in machine learning effectiveness

A recommender system makes its recommendations based on user features, but usually has an underlying machine learned model consisting of an algorithm and a great deal of other features. For this reason, we perform a second statistical analysis to understand the impact and importance of various factors on the effectiveness of the machine learned (ML) model.

Instead of interaction signals, in this model we consider *effectiveness metrics*, and we seek to understand the effect of different factors of the ML model on effectiveness. The metrics we consider are standard metrics such as accuracy, precision, recall, etc., all based on the ability of the model to correctly predict that a user will interact with a recommended item. For this and the following sections, we primarily focus on accuracy; most results hold across all metrics. Our model is formulated as:

$$\begin{aligned}\text{Effectiveness}_{ijklmn} = {} & \text{Market}_i + \text{Feature}_j + \text{Algorithm}_k + \\ & + \text{Size}_l + \left(\text{Market}_i : \text{Feature}_j\right) + \\ & + \left(\text{Market}_i : \text{Algorithm}_k\right) + \\ & + \left(\text{Feature}_j : \text{Algorithm}_k\right).\end{aligned}$$

Our data includes three ML algorithms—Factorisation Machine, Random Forest, and Naïve Bayes Classifier—with four distinct feature sets. There are four different training set sizes.

Upon fitting this model, we find that the feature and algorithm factors, and their interaction, are the most important to predicting

effectiveness. Table 3 shows the results: the feature, algorithm, and feature:algorithm interaction factors have the three highest effect sizes by ω^2. This is not surprising, as machine learning algorithms are deeply influenced by the feature set used, and changing the particular ML algorithm (e.g., using a random forest in place of a neural network) can dramatically affect the result. However, market comes in as the next most important factor, indicating again that which market a model is used in has a significant impact on its efficacy. Furthermore, the factor representing the interaction between market and feature set has a significant effect as well, indicating that the features to be used may differ depending on the market.

The two statistical analyses reported in this section demonstrate that the market factor plays a central role in user interactions and in the effectiveness of machine learning approaches. Based on these outcomes, we turn now to an investigation of the market effect on training machine learning models for music recommendation.

4 CROSS-MARKET TRAINING

We want to understand the actual effect the market(s) selected for training has on the machine learning model we use to provide recommendations. This will help determine whether we can find a good strategy for training market-specific models, and in general which is the best strategy we can adopt to provide recommendation given a market, whether it is a new or an existing one. Thus, to study the effect and the impact of market in our recommendation scenario, we train our recommendation model according to different strategies, called *policies*.

Our aim is to understand the effect and impact of the market, in particular focusing on two concrete problems:

- Decide, for an existing market, whether it is convenient to use a single machine learning model, or rather consider different policies for each market. That is, given a market for which we have a large set of both user and item interaction features, decide which training set we should use in such scenario, and in particular whether we should maintain the current model (i.e., where data comes from all markets), or move to a model where training data comes from specific selected markets;
- Decide, given a new market for which we do not have any user or item feature, which is the more convenient policy to adopt; in other words, which machine learning setting to adopt in such unlabelled scenario. Given that we want to be able to evaluate the goodness of our approach for launching in a new market, we simulate not having data for one of the markets that we are considering, using a leave-one-out market evaluation setting.

We detail next the policies to be investigated empirically.

4.1 Global policies

This strategy uses a set of training data that comes from all markets in which the service is available. The rationale is that if each market contributes with some data, the model should be general enough to adapt to and capture the heterogeneity of different user behaviours. For this reason, we consider a set of policies that are different variants of one other, and where we select the training data from different markets in different but similar ways.

4.1.1 Global policy. The first strategy consists in training the model using a *balanced* set of data coming from all markets, including the test market. Thus, with one market acting as a test case, and n markets in training, each market contributes exactly $1/n$th of the training instances. This policy can be used as a baseline for training on markets in which we have enough data, because also the test market contributes with $1/n$th of the training instances. Thus, this policy can *not* be used to launch in a new market, since a new market would have no training instances to contribute.

4.1.2 Global-ns policy. A variant of the *Global* policy is called *Global-ns*, which stands for "global not-self". For this policy we train the model on a *balanced* amount of data coming from all other markets. In this case, differently from the *global* policy, we do *not* include the test market itself into the training data. Thus, we use training data coming from every other market. This policy can therefore be used as a baseline for training in new markets.

4.1.3 Global-notbal policy. Yet another variant of the *Global* policy is called *Global-notbal*, which stands for "global not-balanced". In this policy, differently from the previous ones, we train the model using a *not balanced* (but still random) set of data coming from all markets. The rationale of this approach is that markets for which we have more data should contribute with more instances in the training, Note that also in this case we include a subset of data from the test market. This policy can be used as a baseline both for training on existing and new markets. For the latter case, the amount of data for the new market starts from zero, and grows as we acquire data from such market.

4.2 Local policies

We use "local policies" to refer to policies trained from a single market rather than all markets. We detail several local policies.

4.2.1 Self-training policy. The *self* policy trains the model on a subset of data that comes from the same test market. The rationale is that, since users from different markets behave differently, using training data from the same market should remove the heterogeneity effect caused by different markets. In other words, the algorithm should get an advantage derived from the fact that the data is *that* of the market. This policy is useful in practice, both for existing and new markets, but only after gathering enough data from the market. In this paper we consider the new and existing market cases.

4.2.2 Foreign-market policies. We consider several different ways to train a model for one market using data from a foreign market.

Random-other policy. We train the model on the data coming from *a single* other market, different from the test one, randomly selected. This policy gives us the statistical expected value of training on a single other market (different from the test one), and, in principle, it can be also used in practice.

Best-other policy. In this policy, we train the model on the data coming from each possible other market, individually. Then, we rank the models effectiveness score on the test market and we select the market which data is associated with the *most effective*

model. Note that, contrary to previous policies, this approach cannot be used in practice, but rather gives us an upper bound on the effectiveness score of training on data from a single other market.

Worst-other policy. For this policy, we train the model on the data coming from every possible other market, individually. Then, we rank the models effectiveness score on the test market and we select the market which data is associated with the *least effective* model. Note that also this approach cannot be used in practice, but gives us a lower bound on the effectiveness score of training on data from a single other market.

Abroad policy. Finally, we test one heuristic selection policy. We train the model on a subset of users that are not from the test market, but have an affinity with the language spoken in such market. Thus we compute, for each user, a score that tells us the affinity between the user and a given language. This score in computed based on the items from the platform a user interacts with. Then, we sort users by their affinity score with the language spoken in the test market. We select users such that we produce a balanced amount of data for every test market. The rationale is that if a user is from a country x but moves/lives in country y, and x and y have different languages, then the user might interact with items in language x; thus we should recommend her/him items as s/he was part of country y. Note that this policy is both good for new and existing markets.

5 EVALUATION

We start with details of the experimental setup. We then focus on specific results, studying Local versus Global markets, the effect of the training size, a comparison between global policies, and an investigation into individual markets.

5.1 Experimental setup

We investigate the difference between policies by comparing a model trained with policy X to a model trained with policy Y in the same market. For each market, we compute the effectiveness score of policy X minus the effectiveness score of Y. The effectiveness score can be any of the metrics mentioned in Section 3.2, but in practice all results reported here are on the accuracy metric.[2] The resulting difference falls into one of the following three cases:

- *above* zero: policy X is *more* effective than policy Y;
- *below* zero: policy X is *less* effective than policy Y;
- *around* zero: policy X is *as effective as* than policy Y.

The specific machine learning algorithms we use are the three from Section 3.2: Factorisation Machine, Random Forest, and Naïve Bayes. We report results only from the Factorisation Machine, as trends hold across all three. When we do training and testing of any ML algorithm with any training policy, we include a random data selection step. For this reason, we always report the average effectiveness metric score obtained over 10 repetitions or folds.

All experiments are offline: training is done on historical log data, and models are tested on historical log data.

5.2 Local vs Global markets

We first investigate the difference between training with the global policy and a set of local policies that correspond to training the model using another single market. We consider the *self*, *abroad*, *best*, *worst*, and *random* local policies from Section 4. Figure 1 (left) shows for each market (dot on the plot) on the x-axis the policy, and on the y-axis the effectiveness score of the policy minus the effectiveness score of global model. Box-and-whisker plots around the dots show the median and upper/lower quartiles of the differences in effectiveness; when the lower quartile is above zero (or the upper quartile below zero) the difference can be considered significantly different.

Self. As we can see, training the machine learning model on data coming from the test market itself (i.e., the self policy, first column on the left side of the plot) achieves significantly higher effectiveness scores than training on the global policy. In practice, this means that a localised ML model trained on data coming from the same market is more effective than a global model. Remember however that the *self* policy can only be used to deal with existing markets, not to launch in new ones.

Abroad. We see that the abroad policy, which corresponds to training the machine learning model based on language affinities, both with 10k and 20k users (second and third column from the left), has median difference right around zero: it is effective for 50% of the markets, and on average obtains the same effectiveness values as the global policy. There are two possible causes for this: (i) the method we use to compute language affinity, and (ii) that users of the service abroad are not representative of native users. We leave for future work the investigation of this matter. In practice, this can be considered to be a failed attempt to identify a good training market or find a good strategy to launch in new markets.

Best, worst and random policies. If we focus on the three rightmost columns, that is the *best*, *worst*, and *random* policies, we see that the random policy has median around zero, while the best and worst policies are practically always above and below zero respectively. Overall this indicates that if we were able to select the best possible market as a training, then we could obtain higher effectiveness scores than the general policy. Nevertheless, if we select a bad market for training we end up with lower effectiveness scores than the general policy. Overall, this suggests that selecting another market as a training is potentially risky.

Furthermore, remember that in this case the market selection has been obtained with an oracle, after considering all the possible markets as a training, without any indication of a practical policy selection method; thus, in general we do not know (yet) how to identify or select a good candidate training market. We provide further insights on the difficulty of finding and characterising a good candidate market for training in Section **??** below.

Overall, the results from the best, worst and random policies suggest that theoretically a good training market exists for each test market we considered, but the choice might not be trivial and can lead to lower effectiveness values than the global policy. In other words, this analysis provides two insights:

[2] Reported trends hold across metrics unless otherwise noted.

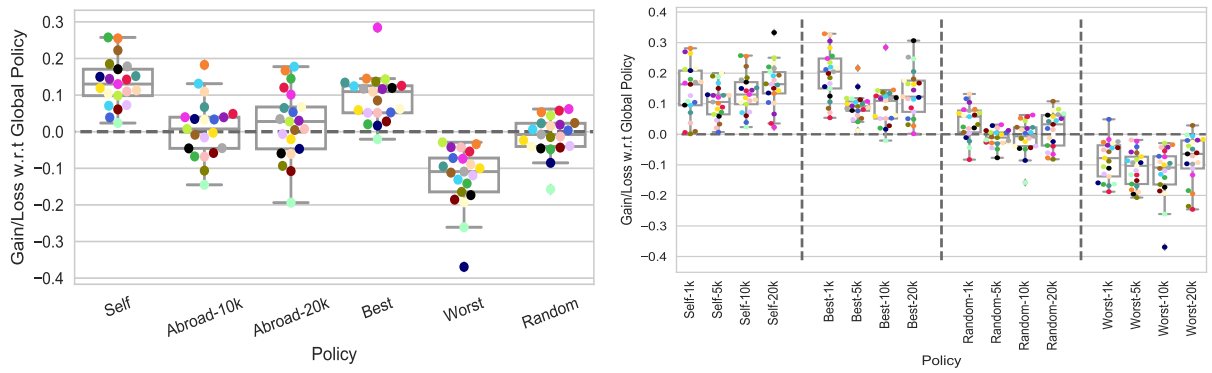

Figure 1: Comparison between the global model and different local training policies. The x-axis specifies a local policy. The y-axis is the difference in accuracy between that policy and the global model. The left plot shows the relative effectivess of the policies, and the right shows the stability of this result over different training set sizes.

(i) moving from a global model to a localised version for existing markets, excluding the case of training with data from the same market (i.e., self policy) is risky; and
(ii) when launching in a new market the safest strategy to adopt is to launch with the global policy rather than trying to identify a possibly good training market.

Conclusions. Overall, we can make the following remarks: training on data from the same market always leads to higher effectiveness values than the global policy, and training on data from another market is a potentially risky approach. We therefore suggest two winning strategies for both new and existing markets. For existing markets we should switch from a global model to a localised version, using training data from the same market (i.e., self policy). With such a strategy we increased the ML effectiveness scores for all existing markets we serve. Concerning new markets, we should launch in such markets with the global policy, and then switch to a localised version after gathering enough data to be able to implement the self policy.

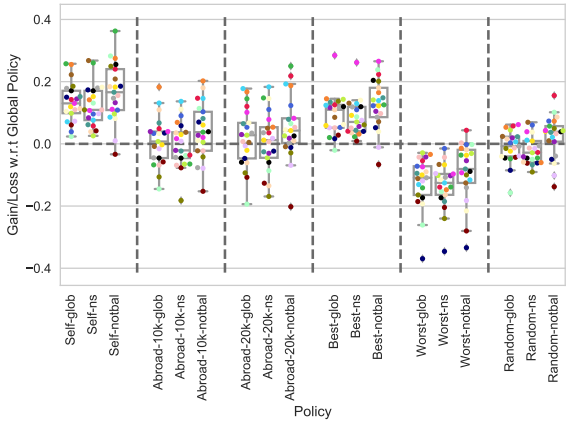

Figure 2: Comparison between different variants of the global policy and local policies. Labels on the x-axis indicate which local policy (self, abroad, best, worst, random) is compared to which global variant (glob, ns, nobal). The y-axis reports different in accuracy between the local model and the global variant.

5.3 Effect of training set size

We next investigate the effect of training set size, and specifically how much data we should have in a market before we switch from a global policy to a localised version. Figure 1 (right) shows the policy on the x-axis, and on the y-axis the effectiveness score of the policy minus the effectiveness score of global model. Each dot is a test market. In this plot, we show how the policies self, best, worst, and random vary across different training data sizes. (The abroad policy has already proven to be ineffective with two different training data sizes.) We consider 1k, 5k, 10k, and 20k training instances randomly selected.[3]

As we can see from Figure 1 (right), all the policies are stable across different training sizes, with 1k items seeming to be an outlier in that it gives unexpectedly better median effectiveness than 5k items. To confirm that it is indeed an outlier, we measure agreement in market ordering using Kendall's τ correlation index. Specifically, we compute the τ rank correlation between two rankings of markets by difference in effectiveness with two different

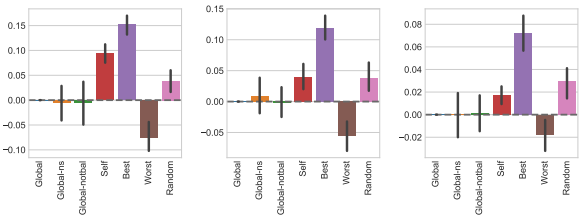

Figure 3: Effect of the different policies on three different markets. Each bar plot represents one market selected at random. The y-axis reports the difference in accuracy between the global model and the corresponding model on the x-axis.

training set sizes. The training size of 5k obtains a higher τ value when compared to 20k rather than 1k has with 20k. This means that

[3]We also did experiments for a subset of markets and policies with 30k, 40k, 50k, and 75k training instances, but results where almost indistinguishable from those with 20k. Thus we selected 20k as the upper bound.

	M16	M4	M19	M1	M18	M17	M14	M10	M8	M13	M5	M6	M15	M11	M20	M7	M12	M3	M2	M9
M	1	0.86	0.85	0.82	0.8	0.63	0.52	0.49	0.43	0.42	0.41	0.33	0.27	0.26	0.26	0.16	0.14	0.01	0.01	0
	0.53	0.53	0.56	0.53	0.54	0.53	0.55	0.55	0.53	0.52	0.54	0.56	0.52	0.55	0.54	0.54	0.54	0.54	0.56	0.55

	M16	M15	M10	M4	M19	M5	M17	M3	M11	M14	M20	M1	M9	M2	M8	M13	M6	M18	M12	M7
M	1	1	0.98	0.93	0.92	0.92	0.92	0.91	0.88	0.86	0.86	0.86	0.69	0.67	0.66	0.58	0.56	0.43	0.32	0
	0.54	0.56	0.55	0.52	0.54	0.54	0.56	0.54	0.56	0.55	0.54	0.57	0.55	0.55	0.55	0.54	0.52	0.55	0.52	0.56

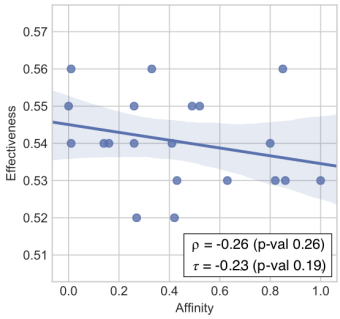

Figure 4: Similarity between pairs of markets and their corresponding effectiveness after training. The left tables compare accuracy values to market similarities for two markets selected at random. The right plot shows the same data as a scatterplot. Note that there is effectively no correlation between accuracy and similarity.

the market ordering of 5k is more similar to the market ordering of 20k, confirming that 1k is not a reliable training size, whereas 5k is.

In practice, this is telling us that, both for new and existing markets, after gathering 5k training instances the more convenient policy to adopt to maximise ML effectiveness is to switch from a global model to a local one. Note that this result gives a sort of guideline for safely launching in a new market: we deploy first the global model, then, after 5k training instances, we switch to the self policy, increasing the recommender system effectiveness.

All previous results have been obtained by comparing any policy with the global one. In the following section, we show that this is not a limitation; all variants of the global policy act similarly.

5.4 Comparison of global policies

So far we have compared the different policies (i.e., self, best, random, etc.) with the global policy. We show now that the different version of global (i.e., Global, Global-ns, and Global-notbal) behave similarly, and thus our previous remarks are sound and independent from the particular global variant we select.

Figure 2 shows the behaviour of the different global policies following the same style as in Figures 1. Each panel of Figure 2 shows a policy (e.g., self), and the three series within each panel representing the difference in effectiveness scores between the given policy and respectively the Global, Global-ns, and Global-notbal policies. For example, the left-most box-plot represents the effectiveness of *self* minus the effectiveness of *global*, the box-plot in the centre represents the effectiveness of *self* minus the effectiveness of *global-ns*, and finally the right most box-plot represents the effectiveness of *self* minus the effectiveness of *global-notbal*.

As we can see from the plot, overall and apart from small fluctuations, the performance of the three versions of the global policy are quite stable. The Global and Global-ns policies are almost indistinguishable, while there are some small (not statistically significant) differences between the Global and Global-notbal policies.

Overall, this shows that all our previous experiments are not dependent on the particular variant of the global policies selected.

5.5 Individual market analysis

We investigate the local vs. global policy decision on a handful of specific market examples. Figure 3 shows, for three representative randomly-sampled markets, bar-plots for the effectiveness of different policies when compared to the global policy. The error bars represent the effectiveness value over training folds. As we can see, confirming our previous remarks, the effectiveness scores of the variants of the global policy are equivalent.

In all cases self-training achieves statistically significantly higher effectiveness scores than any global policy. On the other hand, in several of these cases the best training market gives significantly higher effectiveness than self-training (and in the others no significant difference). From this we conclude that the least risky policy to deploy is self-training, but there is still potential in trying to identify better training candidates in some markets.

5.6 Summary

Taken together, our results strongly suggest a winning and sound strategy to deploy an effective recommender system both in new and existing markets: training models on user data from the market when at least 5,000 training instances are available, and using a globally-trained model before that. Of course, we cannot claim that this exact result generalizes to other recommender systems or domains; whether the number is 5,000, 10,000, or 20,000, the point is that it takes a relatively small number of training instances to fit a model that is good enough for deployment. Nevertheless, we see evidence that being able to identify a good training market could be even more effective than the self policy.

6 FURTHER ANALYSIS

The experiments and results detailed above show that it is hard to beat training a market model based on its own users' data. But if we could identify a good training market, we would be able to further improve our training strategy for both new and existing markets. For this reason, we perform some experiments with the aim of identifying a good training market given a test market.

To do this, we investigate several methods for computing a similarity score between two markets. Given a test market, we train a model on the most similar market identified using several different approaches: the ANOVA analysis on the user factors (from Section 3.1; the ANOVA analysis on the ML factors (Section 3.2); and various combinations of these two.

The tables in Figure 4 (left) show results for two test markets (randomly sampled from the set of 21). The top row of each of the two tables gives the similarity between the test market and each of the 20 candidate training markets (normalized to [0–1]). The bottom row of each table gives the effectiveness when training the model on data from the column market. Figure 4 (right) shows the scatter-plot of the similarity between markets (x-axis) and the effectiveness of using that market for training (y-axis). Finding a correlation in this data would give us reason to believe that there is a way to select a good training market.

As can be seen, however, there is effectively no correlation between our market similarity scores and the resulting effectiveness: the correlations are negative but not statistically significant (see the scatter-plot in Figure 4). This holds regardless of the method for computing similarity (among those tested). In practice this means we obtain effectiveness values in general lower, and very rarely comparable to, the ones obtained with the general policy. This, in conjunction with results above, shows that the problem of finding a good training market is not straightforward, and leave for future work an extensive comparison between market selection strategies.

7 CONCLUSION

Many modern recommender systems are optimised to deliver recommendations to millions of users spread across markets. The challenge is whether to deploy a single global model across all markets, or develop custom models for different markets. In this paper, we provide some answers to this, within the lens of a specific case of music recommendation across 21 different markets. To address RQ1 (whether market is an important attribute to consider when deploying machine-learned models), we presented two statistical analysis that can be used to break down user heterogeneity and allow to leverage such signals. The first statistical analysis breaks down user features and consider user heterogeneity and market effect on user interaction signals. Findings suggest that market is the main effect, and that users behave and interacts differently in different markets. The second analysis shows that while ML algorithms are mainly influenced by the particular algorithm and feature set, market has a strong impact and interacts significantly with the other factors.

For RQ2 (whether it is more convenient to train a single machine learned model globally for use in all markets, or to consider different policies for different markets), our findings show that after a relatively small number of instances (only 5,000 in our particular environment), the less risky and more effective strategy to adopt that increases the machine learning effectiveness is to train the model using data from the market in which we want to provide recommendations.

Finally, with respect to RQ3 (regarding the best policy to adopt to train a machine learned model in a new market for which we do not have any user or item features), the strategy we identified that allows to increase the machine learning effectiveness is to launch in the new market with the global model, and subsequently switch to a localised version when enough data is available—and because the amount of data needed is small, it may not take long to reach this point. Overall, we believe our findings help in the understanding and the design of a more sound end engineered deployment of multiple machine learning models for music recommendation in different markets. It is our belief that these results would generalize to other recommendation services that operate across markets when there are significant differences in user behavior by market. It is likely that this is common, in part because many services offer different catalog content depending on market and have different features or user interfaces (even if only due to localisation) that can substantially impact user behavior.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. [n.d.]. *Context-Aware Recommender Systems*. Springer US, Boston, MA, 217–253.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), 734–749.

[3] David Banks, Paul Over, and Nien-Fan Zhang. 1999. Blind men and elephants: Six approaches to TREC data. *Information Retrieval* 1, 1 (1999), 7–34.

[4] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender systems handbook*. Springer, 919–959.

[5] Benjamin A. Carterette. 2012. Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM TOIS* 30 (2012), 4:1–4:34.

[6] Hung-Chen Chen and Arbee L. P. Chen. 2001. A Music Recommendation System Based on Music Data Grouping and User Interests. In *Proc. CIKM* (Atlanta, Georgia, USA). 231–238.

[7] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proc. WWW* (Florence, Italy). 278–288.

[8] Katayoun Farrahi, Markus Schedl, Andreu Vall, David Hauger, and Marko Tkalcic. 2014. Impact of listening behavior on music recommendation. (2014).

[9] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2011. A Generic Semantic-based Framework for Cross-domain Recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (Chicago, Illinois) *(HetRec '11)*. 25–32.

[10] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish conference on information retrieval*. sn, 24.

[11] Nicola Ferro and Donna Harman. 2009. CLEF 2009: Grid at CLEF pilot track overview. In *Workshop of CLEF*. Springer, 552–565.

[12] Nicola Ferro, Yubin Kim, and Mark Sanderson. 2019. Using Collection Shards to Study Retrieval Performance Effect Sizes. *ACM Trans. Inf. Syst.* 37, 3, Article 30 (March 2019), 40 pages.

[13] Nicola Ferro and Mark Sanderson. 2017. Sub-corpora Impact on System Effectiveness. In *Proc. SIGIR* (Shinjuku, Tokyo, Japan). 901–904.

[14] Nicola Ferro and Gianmaria Silvello. 2016. A General Linear Mixed Models Approach to Study System Component Effects. In *Proc. SIGIR* (Pisa, Italy). 25–34.

[15] Nicola Ferro and Gianmaria Silvello. 2018. Toward an anatomy of IR system component performances. *JASIST* 69, 2 (2018), 187–200.

[16] Bruce Ferwerda, Andreu Vall, Marko Tkalcic, and Markus Schedl. 2016. Exploring Music Diversity Needs Across Countries. In *Proc. UMAP* (Halifax, Nova Scotia, Canada). 287–288.

[17] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 2004. *Bayesian Data Analysis* (2nd ed. ed.). Chapman and Hall/CRC.

[18] Il Im and Alexander Hars. 2007. Does a One-size Recommendation System Fit All? The Effectiveness of Collaborative Filtering Based Recommendation Systems Across Different Domains and Search Modes. *ACM Trans. Inf. Syst.* 26, 1 (2007).

[19] Gaya K. Jayasinghe, William Webber, Mark Sanderson, Lasitha S. Dharmasena, and J. Shane Culpepper. 2015. Statistical Comparisons of Non-deterministic IR Systems Using Two Dimensional Variance. *IPM* 51, 5 (Sept. 2015), 677–694.

[20] Zhongqi Lu, Erheng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang. 2013. Selective transfer learning for cross domain recommendation. In *Proc. SIAM*. SIAM, 641–649.

[21] Peter McCullagh and James A Nelder. 1989. Generalized Linear Models, Vol. 37 of Monographs on Statistics and Applied Probability.

[22] Donald F Morrison. 2005. Multivariate analysis of variance. *Encyclopedia of biostatistics* 5 (2005).

[23] John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society* 135, 3 (1972), 370–384.

[24] Stephen E. Robertson and Evangelos Kanoulas. 2012. On Per-topic Variance in IR Evaluation. In *Proceedings of the 35th International ACM SIGIR* (Portland, Oregon, USA). 891–900.

[25] Andrew Rutherford. 2011. *ANOVA and ANCOVA: a GLM approach*. John Wiley & Sons.

[26] Markus Schedl and David Hauger. 2015. Tailoring Music Recommendations to Users by Considering Diversity, Mainstreaminess, and Novelty. In *Proc. SIGIR* (Santiago, Chile). 947–950.
[27] Jean Tague-Sutcliffe and James Blustein. 1995. A statistical analysis of the TREC-3 data. *NIST SPECIAL PUBLICATION SP* (1995), 385–385.
[28] Fabio Zampieri, Kevin Roitero, J. Shane Culpepper, Oren Kurland, and Stefano Mizzaro. 2019. On Topic Difficulty in IR Evaluation: The Effect of Systems, Corpora, and System Components. In *Proc. SIGIR* (Paris, France) *(SIGIR'19)*. 909–912.
[29] Peng Zhang, Dawei Song, Jun Wang, and Yuexian Hou. 2014. Bias–variance analysis in estimating true query model for information retrieval. *IPM* 50, 1 (2014), 199–217.
[30] Yi Zhang and Jonathan Koren. 2007. Efficient Bayesian Hierarchical User Modeling for Recommendation System. In *Proc. SIGIR* (Amsterdam, The Netherlands) *(SIGIR '07)*. 47–54.
[31] Lili Zhao, Sinno Jialin Pan, Evan Wei Xiang, Erheng Zhong, Zhongqi Lu, and Qiang Yang. 2013. Active transfer learning for cross-system recommendation. In *Proc. AAAI*.