

# Multi-Gradient Descent for Multi-Objective Recommender Systems

Nikola Milojković,<sup>1\*</sup> Diego Antognini,<sup>2</sup> Giancarlo Bergamin,<sup>3</sup> Boi Faltings,<sup>2</sup> Claudiu Musat<sup>3</sup>

<sup>1</sup>Oracle Labs, Switzerland

<sup>2</sup>EPFL, LIA, Switzerland

<sup>3</sup>Swisscom, Switzerland

firstname.lastname@{oracle.com, epfl.ch, swisscom.com}

## Abstract

Recommender systems need to mirror the complexity of the environment they are applied in. The more we know about what might benefit the user, the more objectives the recommender system has. In addition there may be multiple stakeholders - sellers, buyers, shareholders - in addition to legal and ethical constraints. Simultaneously optimizing for a multitude of objectives, correlated and not correlated, having the same scale or not, has proven difficult so far.

We introduce a stochastic multi-gradient descent approach to recommender systems (MGDRec) to solve this problem. We show that this exceeds state-of-the-art methods in traditional objective mixtures, like revenue and recall. Not only that, but through gradient normalization we can combine fundamentally different objectives, having diverse scales, into a single coherent framework. We show that uncorrelated objectives, like the proportion of quality products, can be improved alongside accuracy. Through the use of stochasticity, we avoid the pitfalls of calculating full gradients and provide a clear setting for its applicability.

## Introduction

Today, **recommender systems** are an inevitable part of everyone's daily digital routine. When a person goes online, they are likely going to use one of the services in which recommendation plays an important role. This applies to streaming music, shopping, socializing on social media platforms or viewing a personalized news feed, among many others. The content is tailored to the user during these activities and is selected by the recommender systems. This automated selection of relevant content makes the entire experience of using digital services more comfortable and engaging (Knijnenburg et al., 2012). Without this, the user would be lost in the enormous and continuously growing quantity of information, products, or choices.

To create the best possible user experience - the most useful recommendations, usually, multiple criteria (often conflicting) have to be taken into account. For example, recommending to users the set of  $K$  books, always selected from the set of best-selling books, will not give these users an opportunity to pick some, maybe unpopular, but possibly more

appealing book (Abdollahpouri et al., 2017). Thus, to reduce the popularity bias in this scenario, the **additional criteria** would be to create not only relevant but also diverse recommendations.

Regularly the incentives of the service provider are aligned with the satisfaction of the users of their service. However, the provider is also expecting certain benefits from operating the system. For example, one of the objectives of the online retailer might be to increase their profit. Therefore, during the design of such system we also need to take into account the interests of all parties - stakeholders. These may be correlated but they are not necessarily identical and the differences, however subtle, can give rise to tension in converging towards a generally satisfactory solution.

Both of these, additional criteria and stakeholders, increase the level of complexity with which the designers of the recommender system have to deal. In this paper we present a general multi-objective optimization algorithm which resolves these issues. Unlike (Burke et al., 2018), we go beyond listing the types of possible objectives and propose a way to jointly optimize them.

**Multi-objective recommender systems** have long been impractical due to the heavy computation cost involved in the joint optimization. Traditional approaches include evolutionary and genetic algorithms (Lin et al., 2018; Lin et al., 2019a; Geng et al., 2015). However, these models can only be applied on tiny sets of users and items, which do not scale beyond datasets counting hundreds of samples. This situation does not reflect real use-cases, where we can encounter orders of magnitude more products (Gomez-Urbe and Hunt, 2015) and, similarly, orders of magnitude more users (Linden et al., 2003).

To alleviate computational issues, other methods optimize each objective sequentially. Generally the accuracy is optimized in a first step such as (Di Noia et al., 2017; Jugovac et al., 2017), which leads to an initial ranking of the items for a particular user. In a second step, the items are re-ranked using one or more additional objectives. This shortcoming is more visible in problems where the product space is tightly constrained. When few products are available to choose from, if the objectives are not correlated, the rankings for each objective will be materially different. Too few items that are ranked highly for the first objective will also be good solutions for the remaining ones.

\*Work done while at EPFL and Swisscom.

To obtain the best of both worlds, existing methods employ a weighted average: either in weighting different losses or in weighting different rankings obtained for each criteria. (Ribeiro et al., 2015) propose a method that combines the output of different algorithms trained for different criteria, and aggregates their ranked lists to provide the final recommendation.

Finally, **multi-gradient descent** approaches have been proposed to optimize all objectives simultaneously (Lin et al., 2019b), and provides a better weighting aggregation. In (Lin et al., 2019b) multi-gradient descent is shown to work on correlated objectives. There are however doubts with regard to the mixture of varied objectives and also the applicability on non differentiable functions.

In this work, we address both of these limitations of previous work on multi-objective recommenders based on multi-gradient descent (MGDRec). We first focus on a traditional setting where the optimization is done for two correlated objectives. On two separate datasets, with data on movies and books, we show that MGDRec can create solutions that approach the theoretical optimum - the combination of the best result that can be obtained for each individual objective.

We extend this analysis to show that through gradient normalization and specific training procedures we can extend MGDRec to **non-correlated objectives**. We focused on recommending unpopular items - documentaries in a movie recommender setup - whose proportion in the recommendation set is anticorrelated with recall. Furthermore, we extend the method to non differentiable problems by using the stochastic multi-subgradient descent algorithm (SMSGDA) instead of multi-gradient descent algorithm (MGDA). Unlike previous work - (Lin et al., 2019b), we **formalize the use of stochastic optimization** and provide a clear setting for its applicability.

The main contributions of this paper are:

- We find a set of solutions to multi-objective recommendation problems combining varied objectives, using multi-gradient descent. We show that this yields results superior to the state of the art.
- We introduce the novel idea of gradient normalization to the multi gradient recommendations. This allows us to combine fundamentally different objectives into the same objective function by using subgradients to relax the differentiability conditions for individual objectives. This flexibility allows us to deal with objectives coming from multiple stakeholders.

## Related Work

In recent years a lot of effort in recommender systems research is oriented towards improving end-user experience. This led to the increasing interest in objectives other than accuracy, and consequently to various approaches in the design of multi-objective recommender systems.

### Approaches based on Evolutionary Algorithms

Traditional approaches include evolutionary or genetic algorithms such as Non-dominated Neighbor Immune Algorithm based Recommender System (NNIA-RS) (Geng

et al., 2015), Probabilistic Multi-Objective Evolutionary Algorithm (PMOEA) (Cui et al., 2017), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002), Decomposition-based Multi-Objective Evolutionary Algorithm (MOEA/Ds) (Lin et al., 2019a) or Multi-Objective Evolutionary Algorithm with Extreme Point Guided (MOEA-EPG) (Lin et al., 2018).

The main disadvantage of this kind of approach is bad scalability. For example, in (Lin et al., 2018), the proposed method grows quadratically with the number of users and linearly with the number of items, which leads to a high computational complexity. Therefore, these models can only be applied on tiny sets of users and items, which do not scale beyond datasets counting hundreds of samples.

### Re-Ranking

To avoid scalability issues, other works proposed a setup where the recommender is optimized for relevance objective, and then the additional objective is being used for re-ranking. Examples of such methods are Multiple Objective Optimization in Recommendation Systems (Rodriguez et al., 2012), the greedy strategies of (Di Noia et al., 2017) using Maximal Marginal Relevance (MMR) (Vargas and Castells, 2011), Explicit Query Aspect Diversification (xQuAD) (Vargas and Castells, 2013), or user-interaction based (Pu and Faltings, 2000; Faltings et al., 2004), and the post-processing method Personalized Ranking Adaptation (PRA) (Jugovac et al., 2017). Also, there is substantial work where the solution is proposed only for a specific objective.

However, these methods apply re-ranking either 1) in a post-processing manner (Jugovac et al., 2017) or 2) on the top- $n$  recommended items during the optimization process (Di Noia et al., 2017). The former solution leads to sub-optimal solutions, because the recommendation has been trained on a single objective without taking into account others. The latter suffers, in addition, from popularity bias (Abdollahpouri et al., 2017), as only the most likely items for a user are reordered.

### Weighted-Sum of Objectives

The intuitive and seemingly easy solution to scalability and sub-optimal solutions is to transform the multi-objective problem into a single objective problem. This new single objective would be a weighted sum of all objectives. (Ribeiro et al., 2015) aggregate multiple ranked lists, weights from a graph modeled under a constraint satisfaction problem framework (Torrens and Faltings, 2002). (Lin et al., 2019b) use gradient-descent approaches to provide a better weighting aggregation. Nevertheless, gradient-based methods cannot be applied with non differentiable functions. Finally, the main issue with these method is how to pick an optimal set of weights. In real-world problems doing a grid-search to find these weights might be extremely expensive.

### Problem Formulation

In this section we formulate the problem of Multi-Objective Optimization, and define its solution with the Pareto Optimal Solution.

## Multi-Objective Optimization Problem

A multi-objective optimization problem (MOOP) is an optimization problem in which several possibly conflicting objectives are being optimized simultaneously. It can be defined as follows:

$$\min_{w \in R^D} \mathbf{L}(w) = \min_{w \in R^D} \begin{pmatrix} L_1(w) \\ L_2(w) \\ \vdots \\ L_n(w) \end{pmatrix}_{n \times 1} \quad (1)$$

where  $n$  is the number of objectives to optimize,  $w$  the model parameters,  $D$  the total number of parameter,  $L_i : R^D \rightarrow R, i = 1, \dots, n$ ,  $L_i$  is a single objective loss function, and  $\mathbf{L}$  the multi-objective loss function. In case that a gradient-based optimization algorithm is applied, the gradient of every constituent loss function  $\nabla_w L_i(w)$  has to be a Lipschitz continuous function (Murphy, 2013).

The operator *min* in Equation 1 represents the operation of minimization of all objectives simultaneously. This is not a limiting factor, because, without loss of generality, any maximization problem can be transformed into a minimization problem.

### Pareto Optimal Solution

Unlike the single-objective optimization problems, in multi-objective optimization problems, in general, there will not exist a unique solution that is better with respect to all objectives. This holds as we cannot make any assumption about the relationship of constituent objectives (whether they are correlated, not correlated, linearly dependent, or independent). Thus, the solution to the MOOP is not a single solution, but a set of solutions; this set of solutions is called Pareto Set. Before formally defining a concept of Pareto Optimal solution, we first define the concept of Pareto Dominance.

**Definition 1:** A solution  $w^*$  **dominates** solution  $w$  if for all objectives  $L_i(w^*) \leq L_i(w), i = 1, \dots, n$  and at least for one objective  $L_j(w^*) < L_j(w), j = 1, \dots, n$ .

**Definition 2:** A solution  $w^*$  is **Pareto Optimal** if it is **not dominated** by any other solution  $w$ .

**Definition 3:** The set of all **non-dominated** solutions is called **Pareto Set**.

### Types of Objectives

One of the fundamental goals of recommender systems is to create relevant recommendations. Naturally, there is no use in recommending irrelevant items. However, in real-world applications, there are other objectives that we need to satisfy besides relevance: "Good businesses pay attention to what their customers have to say. But what customers ask for and what actually works are very different." (Gomez-Uribe and Hunt, 2015). In this section, we describe different types of objectives that occurs in recommender systems domain.

## Semantic Relevance

Semantic relevance, relevance, accuracy, correctness, all of these are different names denoting the same goal of recommending a just-right item to the end-user. In other words, the goal is to recommend an item or a set of items which the end-user is most probably going to like. A tremendous amount of time and effort are spent both in research and industry towards creating new recommender systems that are better than the state-of-the-art with respect to accuracy. Definitely, this is the most important objective in every recommender system. However, it is not the only one we should care about, because the end-user satisfaction is not always correlated to the relevance (McNee et al., 2006).

### Correlated to Semantic Relevance

This class of objectives are correlated to the semantic relevance objective. For example, it is not possible to have a *Revenue* bigger than zero if you recommend just irrelevant items. Thus, the *Revenue* is the example of an objective that is correlated with the semantic relevance objective.

### Not Correlated to Semantic Relevance

Finally, this type of objectives is those that are not correlated to the semantic relevance objective. Contrary to the correlated objectives, here we may have a perfectly *fair* recommendations (if *Fairness* is, for example, an additional objective), and yet these recommendations can be completely irrelevant.

## Optimization Algorithm

To solve the multi-objective optimization problem we will use a gradient-based optimization algorithm. Before presenting the algorithm, we introduce the Common Descent Vectors and present the conditions of optimality for gradient-based solutions in MOOP.

### Common Descent Vector

A common descent vector is a convex combination of gradients of each objective. It can be defined as follows (Désidéri, 2012):

$$\nabla_w \mathbf{L}(w) = \sum_{i=1}^n \alpha_i \nabla_w L_i(w) \quad (2)$$

where  $n$  is number of objectives,  $w$  the model parameters,  $\nabla_w \mathbf{L}(w)$  the common descent vector,  $\nabla_w L_i(w)$  the gradient of the objective function  $i$ ,  $\alpha_i$  weight of the  $i^{th}$  gradient. Equation 2 satisfies the following conditions:

1.  $\alpha_1, \dots, \alpha_n \geq 0$
2.  $\sum_{i=1}^n \alpha_i = 1$

### Optimality Conditions

In a deterministic multi-objective gradient-based optimization, the necessary conditions for a solution to be optimal are the Karush-Kuhn-Tucker (KKT) conditions. Every solution that satisfies these conditions are called **Pareto Stationary** (Désidéri, 2012).

**Definition 4:** A solution  $w$  is said to be **Pareto Stationary** if there exists  $\alpha_1, \dots, \alpha_n$  such that:

1.  $\alpha_1, \dots, \alpha_n \geq 0$
2.  $\sum_{i=1}^n \alpha_i = 1$
3.  $\sum_{i=1}^n \alpha_i \nabla_w L_i(w) = 0$

However, the Pareto Stationarity is only the necessary condition of optimality, but not sufficient. The explanation for this can be found if we start from the single objective case: in a single-objective optimization zero gradient is only a necessary condition. It extends to multi-objective problem where the convex combination of gradients is zero, which is just a necessary condition of optimality, but not sufficient.

**Definition 5:** Every **Pareto optimal** solution has to be **Pareto stationary**, but not every **Pareto stationary** solution is **Pareto optimal**.

### Multi-Gradient Descent Algorithm (MGDA)

The Multi-Gradient Descent Algorithm (MGDA) is an extension of the classical Gradient Descent Algorithm to multiple objectives. This algorithm is proved to converge to the Pareto Stationary solution (Désidéri, 2012).

In (Désidéri, 2012), the author defines the common descent vector as a minimum L2 norm element in the convex hull of the gradients of each objective. Considering this definition, finding the weights in common descent vector can be formulated as the Quadratic Constrained Optimization Problem (QCOP) (Désidéri, 2012).

The QCOP is defined as follows:

$$\min_{\alpha_1, \dots, \alpha_n} \left\{ \left\| \sum_{i=1}^n \alpha_i \nabla_w L_i(w) \right\|^2 \mid \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0 \right\} \quad (3)$$

After solving QCOP, the common descent vector can be calculated and based on that value, and we either have that:

- $\nabla_w \mathbf{L}(\mathbf{w}) = 0$ , the solution is Pareto Stationary;
- $\nabla_w \mathbf{L}(\mathbf{w}) \neq 0$ , the solution is not Pareto Stationary and  $\nabla_w \mathbf{L}(\mathbf{w})$  is the common descent vector for all objectives.

Based on the number of objectives, there are two different ways of how QCOP can be solved: with an analytical solution for two objectives or with a constrained optimization problem for more objectives.

### Two Objectives

In case of two objectives the QCOP can be defined as:

$$\min_{\alpha \in [0,1]} \|\alpha * \nabla_w L_1(w) + (1 - \alpha) * \nabla_w L_2(w)\|^2 \quad (4)$$

Then, there is an analytical solution to this problem:

$$\alpha = \frac{(\nabla_w L_2(w) - \nabla_w L_1(w))^T * \nabla_w L_2(w)}{\|\nabla_w L_1(w) - \nabla_w L_2(w)\|^2} \quad (5)$$

where  $\alpha$  is clipped to  $[0, 1]$ .

### Multiple Objectives

In case of more than two objectives, we cannot compute an exact solution and have to frame the method under a constrained optimization framework. The efficient solution that scales nicely to the high-dimensional problems is proposed in (Sener and Koltun, 2018). The proposed solution is based on Frank-Wolfe constrained optimization algorithm (Frank and Wolfe, 1956). In the experiments presented in this paper, the Frank-Wolfe solver is efficient and has excellent convergence properties. Hence, the impact on performance (training time) is insignificant.

### Stochastic Multi-Subgradient Descent Algorithm (SMSGDA)

Unfortunately, Multi-Gradient Descent Algorithm suffers from multiple drawbacks:

1. Calculating full gradient at every optimization step is computationally expensive;
2. As a deterministic optimization algorithm, it can quickly become stuck at a bad Pareto stationary point; the same way as a full gradient descent algorithm can quickly become stuck at the bad local minimum;
3. The requirement of calculating gradient for the objective function restricts from using non-smooth loss functions as objective functions (e.g. Mean Absolute Error (MAE)).

These drawbacks limits MGDA from usage in many real-world problems. (Poirion et al., 2017) propose an extension of MGDA to address its limitation, called Stochastic Multi-Subgradient Descent Algorithm (SMSGDA).

The stochasticity in SMSGDA: addresses both computational cost and decreases the probability of being stuck at a bad Pareto stationary point. However, given that we do not calculate full gradient anymore, we are not able to satisfy the third KKT condition in Definition 4:  $\sum_{i=1}^n \alpha_i \nabla_w L_i(w) = 0$ . Therefore, we cannot choose the same stopping criteria anymore. Nevertheless, we can use some of the criteria that are regularly used in stochastic optimization problems. For example, we could stop the optimization process if:

- a number of epochs has been reached;
- the loss of the common descent vector  $\mathbf{L}(\mathbf{w})$  is plateauing;
- the gradient norm is less than  $\epsilon$ .

In (Poirion et al., 2017), the authors prove that SMSGDA almost surely converges if non-smooth loss functions are used as objective functions. This alleviates the third drawback of MGDA.

### Gradient Normalization

When designing recommender systems, we face with various objectives (as described in the Section ). Additionally, these objectives might have values of the different scales. Both MGDA and SMSGDA are sensitive to different value ranges in objective functions; the gradients would have significantly different norms, leading to the case that one objective completely dominates the whole optimization process.

To alleviate the value range in the multiple objective functions, we propose the following gradient normalization method:

$$\nabla_w \hat{L}_i(w) = \frac{\nabla_w L_i(w)}{L_i(w_{init})} \quad (6)$$

where  $\nabla_w \hat{L}_i(w)$  is the normalized gradient vector of a single constituent objective,  $\nabla_w L_i(w)$  the non-normalized gradient vector of a single constituent objective,  $L_i(w_{init})$  the initial loss for the particular objective,  $w$  the model parameters, and  $w_{init}$  the initial parameters of the model. We consider  $L_i(w_{init})$  to be an empirical maximum loss for the particular objective. Consequently, the proposed normalization should have the same effect as having an objective loss function that is almost always in the domain  $L_i(w) \in [0, 1]$ .

### Solution selection

Selecting a Pareto Optimal solution from a Pareto Set is not a trivial task as there is not solution strictly dominating other solutions. However, as a remedy to this problem, there exist several strategies for this task. An overview of the strategies to pick the best solution from a Pareto Set are detailed in (Wang and Rangaiah, 2017). In this paper, we employ the Linear Programming Technique for Multidimensional Analysis of Preference (LINMAP); LINMAP selects an optimal solution based on the Euclidean distance from the ideal point, selecting the solution with the shortest distance (Srinivasan and Shocker, 1973).

## Experiments

### Datasets

In order to assess the effectiveness of our proposed model, we first carried out experiments on the well-known *Movielens 20M* dataset<sup>1</sup>, and on the *Amazon Books* dataset (McAuley and Leskovec, 2013). The second dataset contains multiple target variables that could be used in a multi-objective task. However, the first only includes ratings. We enriched the first dataset by mapping the movies to the *Amazon Movies* dataset (McAuley and Leskovec, 2013), and then extracting the prices. Finally, the target variables are ratings, prices and genres for the *Movielens* dataset, and ratings and prices for the *Amazon Books*.

For both datasets, we employed the same preprocessing procedure: we first binarized 5-star rating (i.e. ratings at three and above are labeled as positive and the rest as negative); users and items were then filtered to those with at least 5 ratings. A summary of the data we used is shown in Table 1. Finally, we divided the data into training, validation, test sets corresponding to a split of 90%, 5%, 5% respectively. Additionally, we masked out 20% of the items for the validation and test sets.

### Objectives

In our experiments we focused on two completely different combinations of objectives, the first one was a combination of semantic relevance and revenue, while the second one was a combination of semantic relevance and content quality.

Datasets	#Users	#Items	#Interactions	Sparsity
Amazon Books	93 976	25 896	964 363	99.9604
Movielens *	132 580	8 936	6 316 389	99.4669

\* Combined with prices from Amazon Movies.

Table 1: Dataset statistics (number of users; number of items; number of user-item interactions; sparsity).

To assess the performance of our model for the semantic relevance objective we use the *Recall@k* metric to measure the ratio of relevant items that are in the top-k recommendations.

$$Recall@k(u, \omega) := \frac{\sum_{r=1}^k \mathbb{I}[\omega(r) \in I_u]}{\min(k, |I_u|)} \quad (7)$$

where  $\omega(r)$  denotes the item at rank  $r$ ,  $I_u$  is the set of held-out items that user  $u$  interacted with and  $\mathbb{I}[\cdot]$  is the indicator function. The denominator is the minimum of  $k$  and the size of the held-out set  $I_u$  (Liang et al., 2018).

The main goal behind the revenue objective is more or less self explanatory: it is a revenue maximization. The metric we use to measure the performance for this objective is *Revenue@k*. That one is a natural extension of the *Recall@k* metric; the added component here is the price of a particular item. More precisely, we compute the mean revenue of the top-k relevant recommended items.

$$Revenue@k(u, \omega, p) := \frac{\sum_{r=1}^k p(r) * \mathbb{I}[\omega(r) \in I_u]}{\min(k, |I_u|)} \quad (8)$$

where  $p(r)$  is the price of the item at rank  $r$ .

Finally, the last objective in our experiments is something that we call content quality. For this one, it is almost impossible to find an universally accepted definition. However, in this context, we refer to content quality as to an objective in which the goal is to increase the amount of “content that matters” in the top-k recommendations. The main motivation for choosing that objective is due to the phenomenon known under the name of “Filter Bubble” (Nguyen et al., 2014). The impact of Filter Bubbles on end-users is that it limits their ability to explore and perceive different content. Who, as a consequence, has a skewed image of the environment. Tackling such a problem requires a lot of effort and interdisciplinary research. However, as a small step forward, we decided to explore the possibility of recommending the “content that matters”. Once again, selecting that kind of content is beyond our knowledge. Thus, we decided to name the documentary genre as “content that matters” (i.e. content quality). Our goal here was to increase the number of documentaries inside the top-k recommendations, which was at the same time also the metric we used for this objective.

### Experimental setup

The approach we propose in this paper is model-agnostic. Therefore the designer of the recommender system has full flexibility in choosing what model he wants to use. For our experiments, we decided to use the Multi-VAE model proposed in (Liang et al., 2018). The architecture of our model

<sup>1</sup><https://grouplens.org/datasets/movielens/20m/>

is identical to the one in that paper. The loss function we use for semantic relevance objective is also unchanged.

For the revenue objective we weight the reconstruction loss of the VAE with a price vector which contains the individual prices of each item.

$$L_{revenue}(w) = price * L_{reconstruction}(w) \quad (9)$$

The intuition behind this loss function is that we want to penalize our model based on the item price. In other words, we want to penalize errors on expensive items more severely.

The loss function for the quality content objective we define as follows:

$$L_{content}(w) = \mathbb{I}_{doc} * popularity * L_{reconstruction}(w) \quad (10)$$

where  $\mathbb{I}_{doc}$  is an indicator vector that signals if an item is a documentary or not and *popularity* is the vector of the popularity of each item (e.g. a count of how many users have interacted with an item). Here we want to penalize our recommender more for making mistakes on documentaries, especially on the ones that are more popular.

## Training Procedure

The training of the model differs slightly between objectives correlated to semantic relevance and objectives not correlated to semantic relevance.

The training procedure starts by computing the initial loss of the model for all objectives. We consider these losses to be the empirical maximum losses. Following this step, we train our model by computing the gradients of all objectives over the individual batches of the training set, we normalize them by the empirical maximum losses and compute the  $\alpha$  parameters as described in Equation 5. We then use these  $\alpha$  parameters to compute the common descent vector illustrated in Equation 4. And finally, we update the parameters of the model. Algorithm 1 shows the training procedure in pseudo code.

---

### Algorithm 1 SMSGDA with Gradient Normalization

---

```

1: initialize()
2: for  $i \in 1, \dots, n$  do
3:    $empirical\_loss_i = L_i(w)$ 
4: end for
5: for  $epoch \in 1, \dots, M$  do
6:   for  $batch \in 1, \dots, B$  do
7:      $do\_forward\_pass()$ 
8:      $evaluate\_model()$ 
9:      $update\_pareto\_set()$ 
10:    for  $i \in 1, \dots, n$  do
11:       $calculate\_loss \ L_i(w)$ 
12:       $calculate\_gradient \ \nabla L_i(w)$ 
13:       $normalize\_gradient \ \nabla L_i(w) = \frac{\nabla_w L_i(w)}{empirical\_loss_i}$ 
14:    end for
15:     $\alpha_1, \dots, \alpha_n = \text{QCOPSolver}(\nabla_w \hat{L}_1(w), \dots, \nabla_w \hat{L}_n(w))$ 
16:     $\nabla_w L(w) = \sum_{i=1}^n \alpha_i \nabla_w \hat{L}_i(w)$ 
17:     $w = w - \eta \nabla_w L(w)$ 
18:  end for
19: end for
```

---

Additional steps we do for the combination of semantic relevance with the non-correlated objectives are that we do not start from the randomly initialized model, but as a starting point, we select the model which was optimized for semantic relevance only. Moreover, in order to nudge the model to learn to recommend items from non-correlated distribution, we inject additional information in end-user preferences. In other words, for the content quality objective, we injected a small positive values for selected content in end-user preference vectors (these values sums up to 1). While this amount is effective for the model to learn the chosen representation, it is not too large to degrade the performance of semantic relevance. Finally, in order to prevent the model from learning immediately to recommend quality content only, the values of  $\alpha$  parameters should be constrained.

## Results

Figure 1 shows the results for the Movielens dataset enriched with price information from the Amazon Movies dataset. MGD with gradient normalization yields the best results in regard to the solution selection we described above. Figure 2 shows a similar result with MDG with gradient normalization outperforming the other approaches.

To assess how well our algorithm performs we compare it to the other well-known approaches to this problem. As a baseline we use the closest solution to the presented one which was proposed by (Rodriguez et al., 2012) (baseline), which is a solution based on re-ranking. In addition we compare our algorithm also to a simple weighted-sum of objectives in versions with and without gradient normalization (“WS w/ GN” and “WS w/o GN”). Finally, we compare it also to the models optimized for the single objectives only (“SRO” for semantic relevance only and “RO” for revenue only).

Both Figures 1 and 2 also show that pure MGD without gradient normalization does not perform well, yielding results comparable to the results of the single objective optimization for semantic relevance. As shown in Figure 2 the results of the approaches with gradient normalization outperform the results of the single objective for revenue on the Revenue@k metric. Note that the weighted sum approach also performed significantly better with gradient normalization than without it. This confirms that gradient normalization plays a vital part in increasing the performance of multi objective optimization algorithms.

Figure 3 shows the results of the experiment with the two objectives semantic relevance and content quality. In this experiment we considered movies belonging to the genre “documentaries” as quality content. The MGD with gradient normalization dominated the other approaches as well. Interestingly, this approach outperformed also the single objective semantic relevance algorithm on this very objective itself. This may be due to additional regularization provided by the quality content objective.

This result shows that for the cost of a small decrease in semantic relevance we can increase the amount of “quality content” in our recommendations drastically.

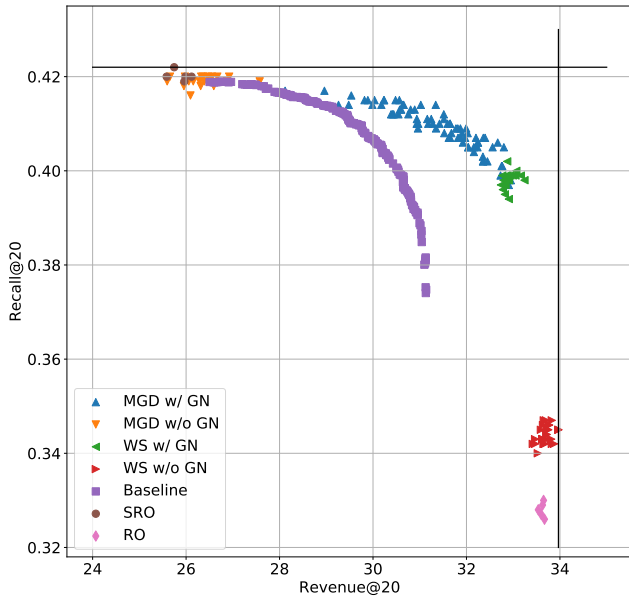


Figure 1: Results for two objectives: semantic relevance and revenue, on MovieLens dataset combined with Amazon Movies dataset.

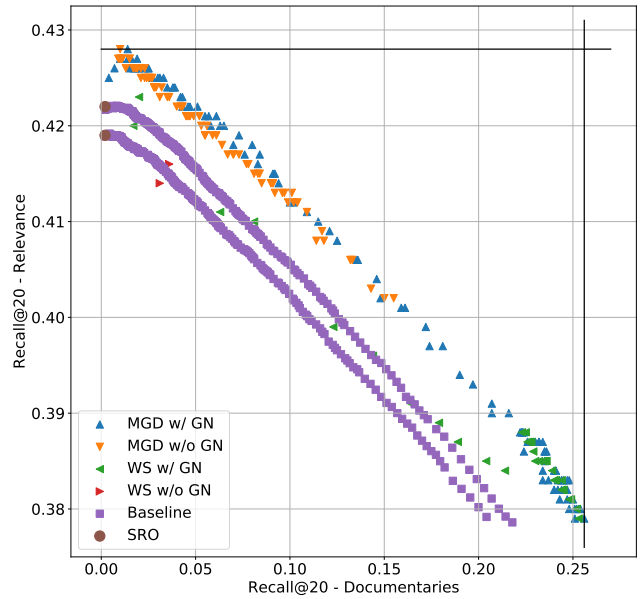


Figure 3: Results for experiments with two objectives semantic: semantic relevance and content quality, on the MovieLens dataset combined with Amazon Movies dataset.

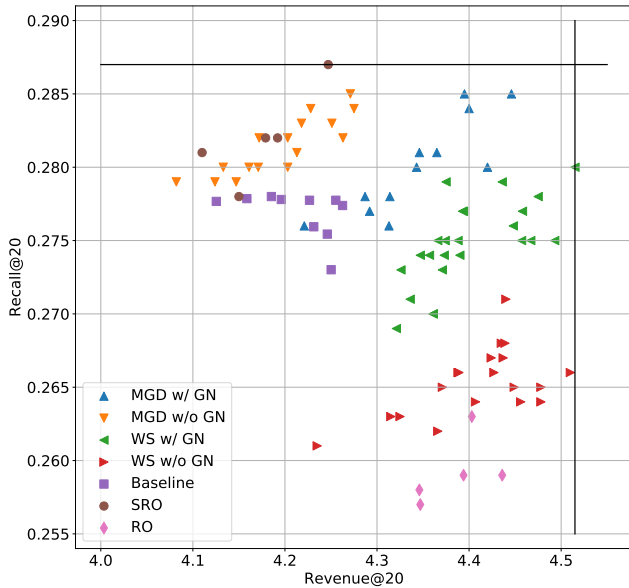


Figure 2: Results for two objectives: semantic relevance and revenue, on Amazon Books dataset.

## Conclusion

The need to optimize for different objectives simultaneously in a recommender system setting is a well recognized problem. While tackling correlated objectives has received more attention previously, it is important for new methods to extend the reach of recommenders to uncorrelated objectives. The optimization problem is more complex when the objectives cease to be correlated or are inversely correlated. In ad-

dition, the various objectives may have different scales and may not be differentiable. For these cases, which are actually the norm and stand for real-world concepts like fairness, diversity or revenue, there is a need for novel methods.

In this work we have shown that multi-gradient descent is applicable in this difficult environment. We first tested in a more traditional setup and showed that revenue and recall can be jointly optimized. In two separate experiments, targeting books and movies, we showed that recommenders based on multi-gradient descent (MGDRec) become the new state of the art. We then show that completely uncorrelated objectives, like the proportion of a certain type of content - for instance quality or unpopular products - can just as easily be brought into the mix.

We solve the problem of the differences of scale between the objectives using normalization techniques - a novelty that is key for getting the state of the art results. Results show that using the gradient normalization leads to solutions that are the closest to the theoretical optimum - the intersection of the best possible value for each objective taken individually.

## References

- Abdollahpouri, H., Burke, R., and Mobasher, B. (2017). Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 42–46. ACM.
- Burke, R., Sonboli, N., and Ordonez-Gauger, A. (2018). Balanced neighborhoods for multi-sided fairness in recommendation. In *Conference on Fairness, Accountability and Transparency*, pages 202–214.
- Cui, L., Ou, P., Fu, X., Wen, Z., and Lu, N. (2017). A novel

- multi-objective evolutionary algorithm for recommendation systems. *Journal of Parallel and Distributed Computing*, 103:53–63.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197.
- Désidéri, J.-A. (2012). Multiple-gradient descent algorithm (MGDA) for multiobjective optimization / Algorithme de descente à gradients multiples pour l’optimisation multiobjectif. *Comptes Rendus Mathématique*, Tome 350(Fascicule 5-6):313–318.
- Di Noia, T., Rosati, J., Tomeo, P., and Di Sciascio, E. (2017). Adaptive multi-attribute diversity for recommender systems. *Information Sciences*, 382:234–253.
- Faltings, B., Torrens, M., and Pu, P. (2004). Solution generation with qualitative models of preferences. *Computational Intelligence*, 20(2):246–263.
- Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.
- Geng, B., Li, L., Jiao, L., Gong, M., Cai, Q., and Wu, Y. (2015). Nnia-rs: A multi-objective optimization based recommender system. *Physica A: Statistical Mechanics and its Applications*, 424:383–397.
- Gomez-Urbe, C. A. and Hunt, N. (2015). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4):13:1–13:19.
- Jugovac, M., Jannach, D., and Lerche, L. (2017). Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, 81:321–331.
- Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., and Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4):441–504.
- Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698. International World Wide Web Conferences Steering Committee.
- Lin, Q., Wang, X., Hu, B., Ma, L., Chen, F., Li, J., and Coello Coello, C. A. (2018). Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation. *Complexity*, 2018.
- Lin, W., Lin, Q., Zhu, Z., Li, J., Chen, J., and Ming, Z. (2019a). Evolutionary search with multiple utopian reference points in decomposition-based multiobjective optimization. *Complexity*, 2019.
- Lin, X., Chen, H., Pei, C., Sun, F., Xiao, X., Sun, H., Zhang, Y., Ou, W., and Jiang, P. (2019b). A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys ’19, pages 20–28.
- Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.
- McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.
- McNee, S., Riedl, J., and Konstan, J. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. pages 1097–1101.
- Murphy, K. P. (2013). *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.].
- Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L., and Konstan, J. A. (2014). Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686. ACM.
- Poirion, F., Mercier, Q., and Desideri, J.-A. (2017). Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications*, 68(2):317–331.
- Pu, P. and Faltings, B. (2000). Enriching buyers’ experiences: the smartclient approach. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 289–296. ACM.
- Ribeiro, M. T., Ziviani, N., Moura, E. S. D., Hata, I., Lacerda, A., and Veloso, A. (2015). Multiobjective pareto-efficient approaches for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*.
- Rodriguez, M., Posse, C., and Zhang, E. (2012). Multiple objective optimization in recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys ’12, pages 11–18. ACM.
- Sener, O. and Koltun, V. (2018). Multi-task learning as multi-objective optimization. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 527–538. Curran Associates, Inc.
- Srinivasan, V. and Shocker, A. D. (1973). Linear programming techniques for multidimensional analysis of preferences. *Psychometrika*, 38(3):337–369.
- Torrens, M. and Faltings, B. (2002). Using soft cpsps for approximating pareto-optimal solution sets. In Junker, U., editor, *AAAI Workshop on Preferences in Constraint Satisfaction (Technical Report WS-02-13)*, pages 99–106. AAAI.
- Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys ’11, pages 109–116. ACM.
- Vargas, S. and Castells, P. (2013). Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th conference on open research areas in information retrieval*, pages 129–136. Le Centre De Hautes Etudes Internationales D’informatique Documentaire.
- Wang, Z. and Rangaiah, G. P. (2017). Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multiobjective optimization. *Industrial & Engineering Chemistry Research*, 56(2):560–574.