Synergy of stochastic and nature inspired optimization in recommender systems

Stepan Balcar, Michal Kopecky, Peter Vojtas[1]

Abstract. In this extended abstract we describe experiences in trying to combine and/or coordinate results of two or more recommendation methods to produce a joint effect greater than the sum of their separate effects - i.e. synergize. We present an overview of results in several directions, lessons learned and some experiments which will guide future research. Our main tool is a dynamic island platform with different stochastic and evolutionary optimization algorithms with migration managed by a multi-agent planner. We present initial results on *MovieLens* datasets.

1 Motivation

To introduce this extended abstract we start with some motivation and overview of previous results.

We focus on the *domain* of recommender systems. Starting from a user-item rating matrix we tried (off-line) to learn user preferences.

In [8] we considered multi-criteria recommendation based on implicit user behavior production data.

Our experimental tool grew up from previous research in domain of artificial intelligence. There we developed the idea of *dynamic island models*, taking inspiration also from multi-agent systems, especially from their mutual help, which can cause synergy in performance. We use the idea of dynamic island models to obtain mutual enrichment of classical stochastic optimization methods and of evolutionary computing in recommender systems. According to [9], first mention of island model is in [10]. Here, already in 1964, S. Wright proposed a basic island model which maintains one subpopulation in one island which is implemented in one cluster so as to achieve parallel evolution. In this model, each cluster carries out own evolutionary algorithm, with frequent interchange of information among them via migration of individuals from one cluster to the others.

Our dynamic island experimental tool was developed in [1]. Experiments on classical combinatorial optimization problems of AI, like TSP, BP etc. were conducted in [4] and [5].

2 Model, experimental prototype, data, metrics, ...

Main idea of PhD theme of the first author was to use this tool for recommendation.

First decision was to find fixed data format for islands and migration.

To be able to implement a dynamic island systems we have to have models that can be exchanged between different optimization algorithms. In recommendation we have chosen latent factors from matrix factorization. These consist of user and item latent factors.

For modeling an individual we use a concatenation of latent factors of users and items. Matrix product of these factors is our estimation of the rating matrix.

In this paper we present a multi-agent tool/system which is available at GitHub[2]. All experiments are repeatable.

First obstacle is the size of models (individuals) which is substantially bigger than that of AI benchmarks, see [3]. Initial experiments are described in [2].

We used data from the movie recommendation domain in the experiments. The effectiveness of parallelization has been verified on *MovieLens* datasets especially on ml-10m[3] datasets, but also on its smaller ml-1m[4] and ml-100k[5] variants.

So far we experiment with RMSE metric.

Our system can be understood as realization of self-played co-evolution and also as a generalized dynamic island model (where on each island a different stochastic/evolutionary optimization can run). The whole is managed by a planner which decides which move to apply in the competition based upon the results of the training samples, motivated also by [7] and [6]. So far our planners are handcrafted.

We have implemented several stochastic and evolutionary optimization methods for collaborative filtering. We compare three types of methods. First, results obtained by running each single methods. Second, results of parallel run of methods when on each island the same method is running (with possibly different parameters). Third, results of co-evolution of different methods. More on this in the Appendix.

We have implemented also a method "content" using similarity on content attributes. Planner manages a portfolio of algorithms in a co-evolutionary manner (see illustration in Figure 1). In this figure we can see how the planner (influenced by migration too) switched between different optimization methods and which (in 1200 iterations) contributed most to best individual. This individual (user and item latent factors) produces a matrix product which is our estimation of

[1] all, Charles University, Faculty of Mathematics and Physics, email: Name.Surname@mff.cuni.cz

[2] https://github.com/sbalcar/distributedea
[3] files.grouplens.org/datasets/movielens/ml-10m.zip
[4] files.grouplens.org/datasets/movielens/ml-1m.zip
[5] files.grouplens.org/datasets/movielens/ml-100k.zip

rating matrix (with respect to different metrics).

Our islands may run diverse computational methods, differing not only in parameters but also in the structure of the whole stochastic algorithm. For the choice of optimal methods for the given problem is responsible the central decision maker, the planner, which replaces unsuccessful method by more successful methods during the whole computation. This option is called *aggressive migration*. In future work we plan to experiment with a compromise between pure parallel run (no migration) and this aggressive one. Our system is enough flexible (to be trained) to enable change of policy during iterations, depending on intermediate results.

Our planner measures ratio of improvement from single methods to parallel run on islands and from best parallel to co-evolution on islands.

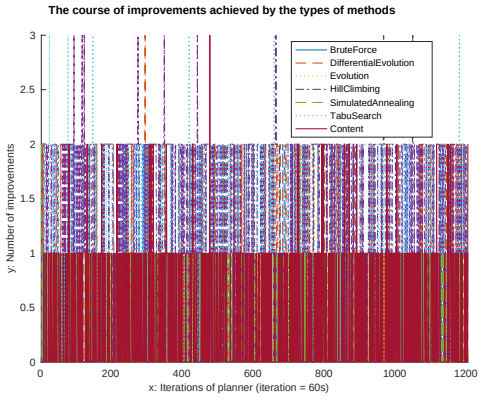

Figure 1. This figure illustrates capabilities of a planner on co-evolutionary island model (just one run), we can see assignments of methods to islands and which of them how many times contributed to the best individual

3 Experiments and results.

Our experiments are two fold.

First we consider regular experiments with training and testing and repetition (cross validation). These are more computationally demanding. In general, parallel run of methods shows improvement in order of 1% when compared to single optimization method in average. Similar improvement (about an additional 1%) we obtain thanks to the effect of co-evolution after 50 iterations in most of decision algorithms managing cooperation and migration between islands. After 1200 iterations the effect is much bigger.

In Appendix we present More details on this. We consider single method compared to two types of island models

- parallel (all islands run the same method and planner and migration can contribute to improvement), and

- co-evolution, when each island runs possibly a different method and all is orchestrated by the planner .

For experiments we use a portfolio of seven stochastic optimization methods. Our testing data are Movie Lens data of three sizes. We show that co-evolutionary island models dominate all others.

Second type of measurements was done with a small randomly chosen representative of data with a bigger variety of ranking metrics (results not depicted here), width of latent factors, number of iterations and a bigger variety of decision makers. Our main target metric is ratio of users with hit in top-10 (in case of ties in highest group). Hierarchical learning of planners is not implemented yet (our experiments are hand crafted). So far we experiment with different representation of planners, history, data. We try to make planners flexible and to adapt to run of computation. Another issue is migration policy. There is much of future work to be done in this direction. First experiments are promising.

For comparison we tried to find the factorization of ml-1m matrix using *Biased Matrix Factorization*, provided by deep neural network. For this comparison we played with different settings and slight modifications of neural network available on GitHub[6]. This NN factorization uses Tensor flow library.

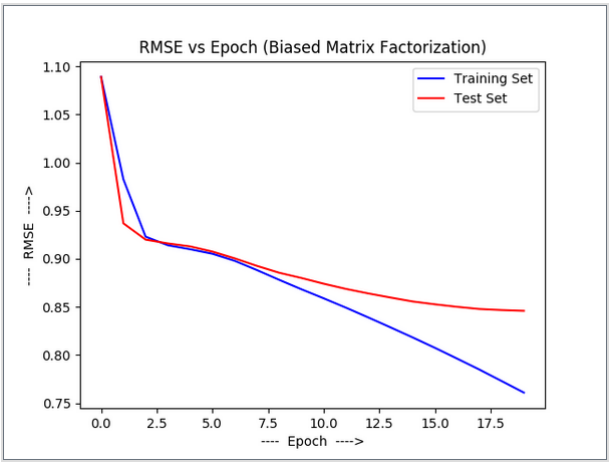

Figure 2. This figure shows initial convergence for [11]'s tool

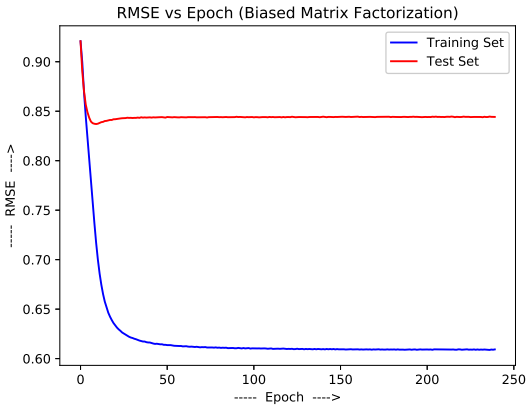

Figure 3. This figure shows our experiments with [11]'s tool. We see possible baseline – RMSE achieved by Biased matrix factorization using Neural Networks, increasing number of learning epochs doesn't improve the quality of the factorization

The graphical representation of original NN convergence of [11] shown on Figure 2[7] presents very rapid descent of RMSE,

[6] https://github.com/zishansami102/Recommendation-Engine

[7] https://github.com/zishansami102/Recommendation-Engine/blob/master/TRvsTS.png

going down to approx. 0.85 on testing data in less than twenty epochs.

Our preliminary tests of the NN behavior show that around this epoch the best results were achieved. Some settings achieved RMSE around 0.835. Further training shows overfitting. While the RMSE on training data continues its descent, the RMSE on testing data starts to grow back to the 0.85 level. Varying top-k (originally 5) considered ratings per user does not substantially change the picture. Only the optimum is moved to higher epochs. It looks promising, because the lost of generalization capability on ml-1m dataset might mean the ability to learn user preferences on bigger datasets.

Our multi-agent system was able to improve baselines but so far cannot compete with deep learning (see Appendix, Table1).

Conclusion - lessons learned and future work. The original approach for small AI problems does not work for substantially bigger problems as recommendation, feature engineering. Main problem is time complexity of operations on big data which makes fast and flexible prototyping impossible. We are considering to send only subsets of models (individuals) during the migration. Sending intermediate results can increase communication frequency and speed up convergence of computation. Processing of incoming migrant should enable enrichment of actual individual (solution designed for AI problems was too aggressive). The research goal is always the same, i.e. to create an independent alone living portfolio of optimization methods for recommendation and also for multi-criteria decision aid. The challenge is to couple our multi-agent system with deep learning.

Acknowledgements. This paper has been supported by Charles University project Progres Q48 and Czech Ph.D grant SVV2018-260451.

REFERENCES

[1] S. Balcar, 'Heterogeneous island model with re-planning of methods (czech)', in *Master thesis, M. Pilát supervisor, https://is.cuni.cz/webapps/zzp/detail/173747/*. Fac. Mathematics and Physics, Charles University, (2017).

[2] S. Balcar, 'Preference learning by matrix factorization on island models', *CEUR Workshop Proceedings*, **2203**, 146–151, (2018).

[3] S. Balcar, 'Technological challenges of distributed agent based parallelization matrix factorization', in *Data A Znalosti & WIKT 2018, http://daz2018.fit.vutbr.cz/DaZ_WIKT_2018_Sbornik.pdf*, pp. 163–168. University of West Bohemia, (2018).

[4] S. Balcar and M. Pilat, 'Heterogeneous island model with replanning of methods', in *GECCO '18*, ed., Hernan Aguirre, New York, NY, USA, (2018). ACM.

[5] S. Balcar and M. Pilat, 'Online parallel portfolio selection with heterogeneous island model (accepted ictai'18)', *http://arxiv.org/abs/1806.04528*, (2018).

[6] E. K. Burke and etal, 'Hyper-heuristics: A survey of the state of the art', *Journal of the Operational Research Society*, **64**(12), 1695–1724, (2013).

[7] C. Jankee, S. Derbel, and C. Fonlupt, 'Distributed adaptive metaheuristic selection', in *Artificial Evolution*, ed., S. Bonnevay et all, pp. 83–96, Cham, (2016). Springer.

[8] L. Peska and P. Vojtas, 'Using implicit preference relations to improve recommender systems', *Journal on Data Semantics*, **6**(1), 15–30, (Mar 2017).

[9] Meng Q, Wu J, Ellis J, and Kennedy P, 'Dynamic island model based on spectral clustering in genetic algorithm', *https://arxiv.org/pdf/1801.01620.pdf*, (2018).

[10] Wright S, 'Stochastic processes in evolution', *Stochastic models in medicine and biology*, **25**, 199–241, (1964).

[11] Zishan Sami, 'Recommendation-engine', *https://github.com/zishansami102/Recommendation-Engine*.

4 Appendix

To be able to measure performance by cooperation of agents we consider also performance of single models. To show that cooperation of agents can have synergistic effects we consider in two levels. In this Appendix we mention experiments to test two hypotheses:

- (H1) First step of cooperation (materialized by exchange of migrants) is between independent runs of same method in parallel on island model. So we hypothesize that parallel islands give better results than single methods,
- (H2) We hypothesize that co-evolution island model give better results than parallel island models. Here we cannot compare method by method, because planners call to cooperation possibly all methods (see 1). So the improvement can be done with respect to best parallel result.

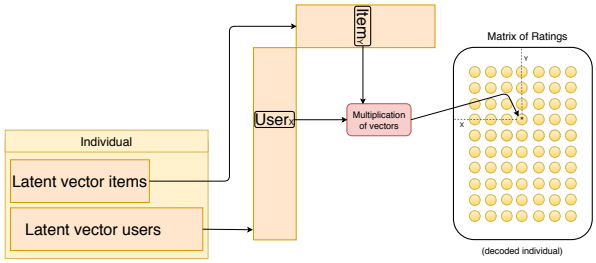

Figure 4. Matrix Factorization and latent factor based individual

To allow interchange of individuals between islands running different algorithms, we needed their uniform representation. We achieved it by concatenating both factor matrix belonging to users and items, shown on figure 4.

First, RMSE results are summarized in Table 1. Columns are organized by datasets. We have chosen three different sizes in order to test the ability of our system. Especially effectiveness of planers on size of an individual. In rows we see three big parts - first are results for single stochastic optimization methods. Second are results for parallel islands (on all islands same method is running). Third part contains results of co-evolutionary islands for different planners. Original hypothesis H1 was that parallel models will give better results than single methods. Further, most important H2, that co-evolutionary island model will give better results than parallel model.

These hypothesis were confirmed only partially. Especially, we can see, that for ml-1M data these are confirmed. Especially for best results obtained by simulated annealing.

For the ml-100k dataset co-evolutionary islands were able to obtain better results than best single method (evolution), and also best parallel islands method (evolution), which failed). For ml-10M dataset in single methods the winner is simulated annealing and was improved in parallel models again by simulated annealing. Surprisingly, co-evolution did not bring improvement on ml-10M dataset (calculated in average values from 9 runs, in best run is co-evolution best the total winner).

Another interesting observation is, that P-BM and P-QI best planners on combinatorial data from [4] and [5] failed here. Probably it is caused by different nature of data. It is not surprising that for combinatorial data other methods are successful than for continuous data (in fact our latent factors are elements of a highly dimensional space and small change in few coordinates factors does influence only few cells in the matrix (product of user and item latent factors) and vanishes in total RMSE.

Methods	ML-100k	ML-1M	ML-10M
si-BruteForce	$0.99149455^{0.99339618}_{0.98967402}$	$1.21664686^{1.24000342}_{1.1910174}$	$1.40924217^{1.43347899}_{1.39578318}$
si-DifferentialEvolution	$1.52094235^{1.52840944}_{1.50868785}$	$1.56321008^{1.57330658}_{1.5553336}$	$1.43739802^{1.44397757}_{1.42759445}$
si-Evolution	$\mathbf{0.88677314}^{0.88981153}_{0.88403419}$	$1.15101941^{1.17967235}_{1.12397231}$	$1.47751841^{1.4931069}_{1.46019633}$
si-HillClimbing	$0.98137083^{0.98412363}_{0.97879639}$	$0.96940294^{0.97061013}_{0.96783751}$	$0.95481835^{0.95558065}_{0.9535425}$
si-RandomSearch	$1.61082133^{1.61502599}_{1.60073478}$	$1.67000129^{1.67349642}_{1.66480817}$	$1.55908779^{1.56414162}_{1.55431825}$
si-SimulatedAnnealing	$1.0119331^{1.02026371}_{1.00519156}$	$\mathbf{0.96909931}^{0.97445829}_{0.96567101}$	$\mathbf{0.91522458}^{0.91724363}_{\mathbf{0.91384568}}$
si-TabuSearch	$0.98044247^{0.98398155}_{0.97819716}$	$0.97068138^{0.97202754}_{0.96973351}$	$0.95449193^{0.95633134}_{0.9534098}$
par-BruteForce	$0.98852673^{0.99183486}_{0.98684871}$	$1.23346875^{1.25793897}_{1.2171644}$	$1.41089326^{1.42969093}_{1.39275724}$
par-DifferentialEvolution	$1.49900561^{1.50711644}_{1.49409959}$	$1.54750813^{1.55401056}_{1.54188967}$	$1.42262659^{1.42670347}_{1.41826442}$
par-Evolution	$\mathbf{0.88759223}^{0.8950173}_{0.87788204}$	$1.12806491^{1.15206596}_{1.10227907}$	$1.4461431^{1.4956173}_{1.42939929}$
par-HillClimbing	$0.97933194^{0.98087785}_{0.97717683}$	$0.96937423^{0.97101474}_{0.9669527}$	$0.95402034^{0.9559042}_{0.95317203}$
par-RandomSearch	$1.60306527^{1.60755238}_{1.59493549}$	$1.66411029^{1.66960367}_{1.65779152}$	$1.55758193^{1.56111558}_{1.55427524}$
par-SimulatedAnnealing	$1.006148^{1.00988576}_{1.00199665}$	$\mathbf{0.96686446}^{0.97004265}_{0.96292779}$	$\mathbf{0.91495706}^{0.91599401}_{\mathbf{0.91397233}}$
par-TabuSearch	$0.9793471^{0.98153915}_{0.97751072}$	$0.96963228^{0.97095041}_{0.96877486}$	$0.95470601^{0.95591173}_{0.95365338}$
coe-StaticWithoutReplanning	$\mathbf{0.8834555}^{0.89124171}_{0.87065507}$	$\mathbf{0.92524342}^{0.93912037}_{0.90587973}$	$0.92051305^{0.9260763}_{0.91050305}$
coe-Random garant (P-RG)	$0.89520003^{0.90484319}_{0.88729023}$	$0.93032483^{0.95760219}_{0.90455501}$	$\mathbf{0.91702632}^{0.93836615}_{\mathbf{0.87978587}}$
coe-Random (P-R)	$0.90036248^{0.93055494}_{0.88507641}$	$0.92908367^{0.95094702}_{0.90721669}$	$0.91703962^{0.94161789}_{0.89183074}$
coe-P-LQI	$0.88698062^{0.88980097}_{0.88480768}$	$0.93337975^{0.95014126}_{0.92753029}$	$0.93698698^{0.94608697}_{0.92842755}$
coe-P-BM	$0.88670551^{0.88886923}_{0.88506827}$	$0.92901411^{0.9329255}_{0.92591107}$	$0.94753561^{0.95153888}_{0.94033183}$
coe-P-QI	$0.88737976^{0.88990467}_{0.88554723}$	$0.93357926^{0.95813448}_{0.91438871}$	$0.93512009^{0.94952515}_{0.92428348}$

Table 1. Results ml-100k, ml-1M, ml-10M - 9 runs, si- prefix means a single method performance, par- prefix means all islands run same method and finally coe- prefix denotes the co-evolution case of our dynamic island model where each island can run a different method tagged by planners names (using terminology of [5]). In red are denoted cases when co-evolution, parallel resp. did not bring improvement