

Constructive Recommendation

Paolo Dragone

University of Trento, Italy
 TIM-SKIL, Trento
 paolo.dragone@unitn.it

Abstract

Constructive recommendation is the task of recommending object “configurations”, i.e. objects that can be assembled from their components on the basis of the user preferences. Examples include: PC configurations, recipes, travel plans, layouts, and other structured objects. Recommended objects are created by maximizing a learned utility function over an exponentially (or even infinitely) large combinatorial space of configurations. The utility function is learned through *preference elicitation*, an interactive process for collecting user feedback about recommended objects. Constructive recommendation brings up a wide range of possible applications as well as many untackled research problems, ranging from the unprecedented complexity of the inference problem to the nontrivial choice of the type of user interaction.

1 Introduction

Research on recommendation system has mainly focused on developing techniques to predict interesting objects from a pool of existing ones (e.g. a database of products). These objects are always “self-contained”, e.g. movies, songs, clothes, and other items that cannot be split into individual components. On the other hand, an aspect that has seldom been investigated is the recommendation of “configurations”, i.e. objects assembled as combinations of individual components and subject to structural constraints. For instance, planning a trip around Italy involves deciding which city visit, in which order, how long stay and which activity do in each city, subject to the available time and budget. Making a decision in this setting involves searching over a combinatorial space of possible candidates. Making good decisions over combinatorial domains is very difficult for humans, and thus computational aid is indeed needed. Recommendation in such structured domains requires the “synthesis” of new objects from scratch (i.e. assemble them from their components) on the basis of the user preferences, a task that we call *constructive recommendation* [Teso *et al.*, 2016; Dragone *et al.*, 2016]. User preferences are represented by a *utility function* $u : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that maps structured objects

Algorithm 1 The Preference Perceptron algorithm.

```

1: Initialize  $\mathbf{w}_1$ 
2: for  $t = 1, \dots, T$  do
3:   Receive context  $x_t$ 
4:    $y_t \leftarrow \operatorname{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}_t, \phi(x_t, y) \rangle$ 
5:   Receive feedback  $\bar{y}_t$ 
6:    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \phi(x_t, \bar{y}_t) - \phi(\mathbf{w}_t, y_t)$ 

```

$y \in \mathcal{Y}$ to a ranking score, which can also depend on contextual information $x \in \mathcal{X}$ (e.g. a user query). Object synthesis is performed by maximizing the utility function over the constrained domain of feasible configurations. The utility function is often assumed to have a parametrized form, such as $u(x, y) = \langle \mathbf{w}, \phi(x, y) \rangle$ where $\mathbf{w} \in \mathbb{R}^d$ is a parameter vector and $\phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ is a d -dimensional feature map. The weight vector \mathbf{w} has to be learned to fit the actual user preferences. Existing recommendation techniques (e.g. collaborative or content-based filtering) are not suited for this task, as they rely on the assumptions that objects can be explicitly enumerated, which is not feasible in the constructive setting since the domain of available objects contains an exponential number of possible configurations. Another shortcoming of existing recommendation techniques is that learning user preferences over large combinatorial domains would require an unfeasible amount of data for each user. A more suitable approach is to estimate \mathbf{w} by iteratively asking user feedback about recommended objects, a process called *preference elicitation*. Preference elicitation [Pigozzi *et al.*, 2016] techniques aim at maximizing the information acquired from each example while minimizing the required interactions and user cognitive effort. State-of-the-art preference elicitation techniques [Viappiani and Boutilier, 2010] are, however, not designed for constructive task either. These methods are based on Bayesian estimation of user preferences, which is too computationally expensive to scale to large constructive domains [Teso *et al.*, 2016]. We instead employ online preference learning techniques, such as Coactive Learning [Shivaswamy and Joachims, 2015], which has already been successfully used in constructive scenarios [Teso *et al.*, 2017; Dragone *et al.*, 2016]. Algorithm 1 shows the Preference Perceptron algorithm from the Coactive Learning framework [Shivaswamy and Joachims, 2015]. At each iteration t , the algorithm receives the context x_t and finds a new recom-

mentation y_t . The user then provides a feedback \bar{y}_t , which is a slightly improved version of y_t with respect to the true utility function of the user (manipulative feedback). The system then uses the pair (y_t, \bar{y}_t) as a ranking example to obtain a better approximation w_{t+1} of the parameter vector through a structured Perceptron update. The new recommendation is found by maximizing the current utility estimate $\langle w_t, \phi(x, y) \rangle$ (Alg. 1, line 4). In constructive recommendation, the set \mathcal{Y} of feasible objects is a constrained combinatorial space and thus the aforementioned maximization problem is a constrained optimization problem. In the simplest case, when \mathcal{Y} is linearly constrained and ϕ is linear in the attributes of y , this problem is a mixed integer linear program (MILP). Despite being NP-hard in the general case, existing MILP solvers can rapidly solve problems with hundreds of variables, making constructive recommendation doable.

2 Research Directions

As summarized in the introduction, constructive recommendation raises several research problems. The following is an outline of some of these problems and the approaches we are currently investigating.

Online Preference Learning

Existing preference elicitation techniques [Viappiani and Boutilier, 2010] cannot scale to large constructive domains [Teso *et al.*, 2016]. Online discriminative techniques, such as Coactive Learning, are better suited to handle the complexity of the underlying configuration domain [Teso *et al.*, 2017; Dragone *et al.*, 2016]. In the constructive setting, however, it is not obvious how the user interaction should be carried out. Manipulative feedback is well suited for some tasks, e.g. layout synthesis [Dragone *et al.*, 2016], but may be difficult to obtain in others, e.g. PC configuration recommendation [Teso *et al.*, 2016] or trip planning [Teso *et al.*, 2017]. Preference elicitation techniques are often based on choice set feedback, which consists in recommending a set of alternatives to the user, who in turn chooses the best one in the set. Choice set feedback is often deemed less cognitively demanding for the user than other types of interactions [Viappiani and Boutilier, 2010]. The only system using choice set feedback designed for constructive recommendation is a large-margin approach from [Teso *et al.*, 2016], which however can work over Boolean-only domains and feature spaces. We are currently working on a Perceptron-based alternative that can handle both Boolean and numeric variables and features. Furthermore, the system from [Teso *et al.*, 2016] comes with no formal guarantees on the recommendation quality, which our approach can instead provide. Preliminary results show improvements on recommendation quality and inference time over the competitor.

Feature Selection and Elicitation

When evaluating complex structured objects, different users may reason over significantly different preference criteria. In general, preference criteria (i.e. features) are arbitrary formulas combining basic objects attributes. Learning over complete feature spaces is often impractical due to the exponential blowup of such combinations. Finding the most relevant

features for the user is a key component of a constructive recommendation system. One possible approach is to elicit features as user feedback, as done in [Teso *et al.*, 2017]. In this work, while using coactive learning to elicit preferences, we also asked the user for “critiques”, i.e. some kind of explanation for dubious examples. Doing so proved to be effective in gradually expanding the feature space with mostly relevant features for the particular user. While being quite effective, this approach requires further user involvement in providing critiques, which may be too cognitively expensive. An alternative that we are exploring is to learn features from preference examples obtained through coactive learning via online gradient boosting [Beygelzimer *et al.*, 2015]. Our aim is to approximate the true user preferences by a linear combination of weak utility learners, which approximate single user features.

Inference over Combinatorial Domains

Recommendations in constructive domains are selected by solving several constrained optimization problems. Especially in the MILP case, existing solvers can handle medium-sized problems reasonably fast. With increasing domain complexity, however, inference becomes impractical. One option to address this issue is to find approximate solutions. While preliminary work has gone in this direction [Dragone *et al.*, 2016], it is still not clear the impact of these approximations in the general case. So far, we have only used approximating heuristics and we still lack a principled approximation technique. Another interesting research question is whether we can go beyond linear programs and address inference in quadratically constrained domains. This would enable us to describe domains involving areas and Euclidean distances, useful e.g. in layout synthesis [Dragone *et al.*, 2016].

References

- [Beygelzimer *et al.*, 2015] Alina Beygelzimer, Elad Hazan, Satyen Kale, and Haipeng Luo. Online gradient boosting. In *Advances in Neural Information Processing Systems*, pages 2458–2466, 2015.
- [Dragone *et al.*, 2016] Paolo Dragone, Luca Erculiani, Maria Teresa Chietera, Stefano Teso, and Andrea Passerini. Constructive layout synthesis via coactive learning. In *CML workshop, NIPS*, 2016.
- [Pigozzi *et al.*, 2016] Gabriella Pigozzi, Alexis Tsoukiàs, and Paolo Viappiani. Preferences in artificial intelligence. *Ann. Math. Artif. Intell.*, 2016.
- [Shivaswamy and Joachims, 2015] Pannaga Shivaswamy and Thorsten Joachims. Coactive Learning. *JAIR*, 2015.
- [Teso *et al.*, 2016] Stefano Teso, Andrea Passerini, and Paolo Viappiani. Constructive preference elicitation by setwise max-margin learning. In *IJCAI*, 2016.
- [Teso *et al.*, 2017] Stefano Teso, Paolo Dragone, and Andrea Passerini. Coactive critiquing: Elicitation of preferences and features. In *AAAI*, 2017.
- [Viappiani and Boutilier, 2010] Paolo Viappiani and Craig Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. In *NIPS*, 2010.