Discovering Personalized Semantics for Soft Attributes in Recommender Systems using Concept Activation Vectors

Christina Göpfert
Bielefeld University
Bielefeld, Germany
chgopfert@gmail.com

Yinlam Chow*
Google Research
Mountain View, CA, USA
yinlamchow@google.com

Chih-wei Hsu
Google Research
Mountain View, CA, USA
cwhsu@google.com

Ivan Vendrov†
Omni Labs
San Francisco, CA, USA
ivendrov@gmail.com

Tyler Lu†
Talka, Inc.
San Francisco, CA, USA
tyler.lu@gmail.com

Deepak Ramachandran
Google Research
Mountain View, CA, USA
ramachandrand@google.com

Craig Boutilier
Google Research
Mountain View, CA, USA
cboutilier@google.com

ABSTRACT

Interactive *recommender systems (RSs)* allow users to express intent, preferences and contexts in a rich fashion, often using natural language. One challenge in using such feedback is *inferring a user's semantic intent* from the open-ended terms used to describe an item, and using it to refine recommendation results. Leveraging *concept activation vectors (CAVs)* [21], we develop a framework to learn a representation that captures the semantics of such attributes and connects them to user preferences and behaviors in RSs. A novel feature of our approach is its ability to distinguish objective and *subjective* attributes and associate *different senses* with different users. Using synthetic and real-world datasets, we show that our CAV representation accurately interprets users' subjective semantics, and can improve recommendations via *interactive critiquing*.

CCS CONCEPTS

• **Information systems → Personalization**.

KEYWORDS

Interactive recommender system, Personalized semantics, Concept activation vectors (CAVs)

ACM Reference Format:
Christina Göpfert, Yinlam Chow, Chih-wei Hsu, Ivan Vendrov, Tyler Lu, Deepak Ramachandran, and Craig Boutilier. 2022. Discovering Personalized Semantics for Soft Attributes in Recommender Systems using Concept Activation Vectors. In *TheWebConf '22: The Web Conference, April 25–29, 2022, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

While *recommender systems (RSs)* have changed how we discover and consume content, products and services, *conversational recommenders* [2] have emerged as a promising paradigm to better understand user needs and preferences—they improve upon the primitive user feedback admitted by traditional RSs (e.g., queries, clicks, item consumption, ratings), allowing users to express their intent, preferences, constraints and contexts in a richer fashion through the use of natural-language-based interaction (e.g., faceted search, dialogue). However, interpreting such interactions requires grounding the user's intended semantics w.r.t. the RS's model of user preferences. For example, if a user expresses a desire for a "funny" movie, this must be translated into an actionable representation of her preferences/intent over the target movie corpus.

When the set of item attributes is well-defined and known *a priori*, existing techniques such as *faceted search* [24, 42] or *example critiquing* [12, 13] can be used directly. But often item attributes are *soft* [1]: there is no "ground truth" association of such soft attributes with items; the attributes themselves may have imprecise interpretations; and they may be *subjective* in nature (i.e., different users may interpret them differently). For instance, in *collaborative filtering (CF)* tasks such as movie recommendation, side information about movie attributes like 'funny,' 'thought-provoking,' or 'violent' is often available, but it is often ancillary, derived from sparse, noisy user comments, reviews, or tags); and, users may disagree on which movies they consider to be 'violent' (or 'too violent').

Recent work has attempted to *jointly* learn the semantics of soft attributes with user preferences [27, 44]. In this work, we adopt a different perspective: we treat the recommendation task as primary, using standard CF models for RSs; and we *infer the semantics of soft attributes using the representation learned by the RS model itself* [14, 36]. This has three advantages: (1) Model capacity is directed to

[*] Contact author.
[†] Work conducted while at Google Research.

TheWebConf '22, April 25–29, 2022, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/1122445.1122456

predicting user-item preferences without side information, which often does not improve RS performance. (2) It offers a means to test whether specific soft attributes are *relevant* to predicting user preferences, and to focus attention on attributes most relevant to capturing of a user's intent (e.g., when explaining recommendations, eliciting preferences, or suggesting critiques). (3) One can learn soft attribute/tag semantics with relatively small amounts of labelled data, in the spirit of pre-training and few-shot learning.

Concretely, we assume we are given: (i) a CF-style model (e.g., probabilistic matrix factorization or dual encoder) which embeds items and users in a latent space based on user-item ratings; and (ii) a set of *tags* (i.e., soft attribute labels) provided by a *subset of users* for a *subset of items*. We develop methods that associate with each item the degree to which it exhibits a soft attribute, by applying *concept activation vectors (CAVs)* [21]—a recent method developed for ML interpretability—to the CF model to detect whether it *learned a representation of the attribute*. The projection of this CAV in embedding space provides a (local) *directional semantics* for the attribute that can then be applied to items. Moreover, the technique can be used to identify the *subjective nature* of an attribute, specifically, whether different users have different meanings (or tag *senses*) in mind when using that tag. Such a *personalized semantics* is vital to the sound interpretation of a user's true preferences.

Our key contributions are as follows: (i) We propose a novel framework using CAVs to identify the semantics of soft attributes relative to preference prediction or behavioral models in RSs *without requiring co-training* of semantics and preference models. (ii) We develop methods to distinguish objective and subjective attributes (both subjectivity of *degree* and of *sense*) and associate different senses of subjective attributes with different users. (iii) We propose a simple method that leverages this semantics to elicit preferences via *example critiquing*. Experiments on both synthetic and real-world data show the efficacy of our methods. Further details and additional experimental results can be found in an extended version of this paper [17].

2 PROBLEM FORMULATION

We first outline our problem formulation then discuss related work.

User-item Ratings. We assume a standard collaborative filtering (CF) task: users $\mathcal{U}$ offer ratings of items $\mathcal{I}$, with $r_{u,i}$ (e.g., 1–5 stars) denoting the rating of user $u \in \mathcal{U}$ for item $i \in \mathcal{I}$. Let $n = |\mathcal{U}|$, $m = |\mathcal{I}|$, and $\mathbf{R}$ denote the $m \times n$ (usually sparse) ratings matrix, with $r_{u,i} = 0$ denoting no rating. Let $R = \{(u,i) : r_{u,i} \neq 0\}$.

Preference Predictions. We assume a CF method has been applied to $\mathbf{R}$ to construct *user and item embeddings*, $\phi_U : \mathcal{U} \mapsto \mathbb{R}^d$ and $\phi_I : \mathcal{I} \mapsto \mathbb{R}^d$, respectively, such that the model's predicted (expected) rating is $\hat{r}_{i,u} = \phi_U(u)^\top \phi_I(i)$. We let $X \subseteq \mathbb{R}^d$ denote the embedding space. Methods include matrix factorization [37] or certain forms of neural CF [3, 45]. For concreteness, we assume a *two-tower model* (or *dual encoder*) in which users and items are passed through separate (but co-trained) deep neural nets (DNNs), N_U and N_I, to produce their respective vector embeddings $\phi_U(u)$ and $\phi_I(i)$, which are combined via dot product to predict user-item affinity $\hat{r}_{i,u}$ [45, 46]. We can view $\phi_I(i)$ as a (learned) latent feature vector characterizing item i and $\phi_U(u)$ as parameterizing user u's estimated *utility (or preference) function* over these features. By construction, user utility is linear w.r.t. these latent item features (a limitation, see below).

Soft Attributes & Tags. CF methods are often used to predict user-item affinity in *content* RSs (movies, music, news, etc.) because *user rating or consumption behavior* is generally far more predictive of user preferences than *hard (known, objective) attributes* (e.g., genre, artist, director) [23]. Despite this, users often *describe* items using *soft attributes* [1], features that are not part of an agreed-upon, formal item specification. For example, movies might be described using terms like 'funny,' 'thought-provoking,' 'violent,' 'cheesy,' etc. We call such terms *tags* rather than attributes, since they are neither applied universally to all items, nor by all users, and users may disagree on their application.[1]

A number of RSs support user-supplied tags [18]. We assume a set of k canonical tags $\mathcal{T}$. We also assume that tags are used "propositionally" (a user chooses to apply a tag or not) though the underlying attributes may be ordinal or cardinal (e.g., a tag 'violent' may refer to some degree of 'violence').[2] *Tag data* comprises a $m \times n \times k$ tensor $\mathbf{T}$ where $t_{u,i,g} = 1$ if user u applies tag g to item i, and 0 otherwise. Let $T = \{(u,i,g) : t_{u,i,g} = 1\}$ and $T_g = \{(u,i) : t_{u,i,g} = 1\}$. Tags are usually strictly sparser than ratings, so we assume $T_g \subseteq R$ for all $g \leq k$. User u may apply multiple tags to the same item. Let $T_u \subseteq \mathcal{I}$ be the set of items tagged by u (using any tag), $T_{u,g}$ those tagged with g, and $T_{u,\overline{g}} = T_u \setminus T_{u,g}$ those tagged by u but not with g. Our tag data is like that used by tag recommenders [16].

Elicitation & Critiquing. CF models, in isolation, are ill-suited to RSs that aim to naturally interact with users to refine knowledge of their preferences. A CF-based RS can actively elicit new ratings at the *item level* [9, 47], but the (uninterpretable) embedding of items does not support *attribute-based interaction*. Tags can help users better navigate the item space. While many preference elicitation and example critiquing methods use *hard attributes*, in content RSs tags often correspond to *soft attributes* and may be *subjective* in nature. If a user requests a "more thought-provoking" movie, the RS's model of the user's preference cannot be updated unless we have a semantics that relates the tag to items.

Concept Activation Vectors. Research on interpretable representations tries to overcome the fact that modern ML models usually learn complex, non-transparent representations of concepts [21, 38]. The *testing CAVs (TCAV)* framework [21] is a one such mechanism that tries to find a correspondence between the "state" of a model (e.g., input features, DNN activation patterns) and human-interpretable concepts. For instance, suppose a DNN has been trained to classify animals in images. Using a small set of images with positive and negative examples of some concept (e.g., "objects with stripes"), TCAV tests whether the DNN has learned a representation of that concept in the form of a vector of activations (CAV) that correlates with its presence. Moreover, using the derivative of the classifier's output w.r.t. the CAV's direction, it measures how important that concept is to its predictions (e.g., how sensitive a "zebra" classification is to the presence of stripes in an image).

[1] Tags may be specified in the RS, or extracted from user descriptions, reviews, etc.

[2] Our techniques can be extended in a straightforward way to Boolean (positive and negative application), ordinal or cardinal tags.

Mapping CAV Notions to RSs. For RSs, we use CAVs to translate between latent item representations learned by a CF model and soft attributes users adopt to describe items and preferences. We briefly detail the adaptation of key CAV concepts to RSs by drawing an analogy between our setting and the image classification setting used to explicate CAVs by Kim et al. [21] (and informally described above). Our DNN CF model $\Phi = (\phi_U, \phi_I)$ is trained on user-item ratings, similar to the multi-class image classifier trained on labeled images. A soft attribute or tag g (say, 'violent') is analogous to a specific image feature (e.g., 'stripes'). We determine if the *item network* N_I has learned a representation of this tag. As in the image setting, where a small set of positive (striped) and negative (non-striped) images is used to identify a CAV, we use a *small* set of positive (tagged) and negative (untagged) items in the same way, though we must account for variability and inconsistency in the tags applied by different users.[3] We refer to Appendix A.1 for a concise list of key concepts and a graphical illustration of how we apply CAVs to RSs (including in example critiquing, see Sec. 5).

Related Work. A number of methods exist for finding the semantics of tags and attributes in RSs using tag data [16, 27] or reviews [28]. While some learn semantics jointly with ratings prediction, others build attribute models "on top of" a ratings prediction model as we do here. Most related to our approach is that of Gantner et al. [16], who learn semantics for tags as a linear combination of latent features (from a BPR model [35]) using k-nearest neighbors or linear regression (see also Cohen et al. [14]). This work is proposed as a means for solving the cold-start problem. Our work differs in its ability to handle nonlinear representations and subjectivity, and its focus on conversational/critiquing RSs. Tag recommendation more broadly [25, 26, 36] bears some connection to our work in modeling the relationship between users, items and tags.

One of our motivations is the use of soft and subjective attributes for critiquing [13], faceted search [42], and preference elicitation in RSs. Radlinski et al. [34] develop a methodology for connecting user preferences with soft attribute usage in conversational RSs. Little work in elicitation for RSs addresses subjectivity, though Boutilier et al. [7, 8] consider "definitional" subjectivity (a very different notion from ours). Subjectivity has been studied in natural language and psycholinguistics, using personalized embeddings [43] and *prototype theory* [32], where subjectivity is related to the similarity of an item to an idealized exemplar.

3 FINDING RELEVANT SOFT ATTRIBUTES

We develop a method for identifying the semantics of *relevant soft attributes* w.r.t. the item embedding representation learned by our CF method. Assume a CF model $\Phi = (\phi_U, \phi_I)$, trained on ratings data $\mathbf{R}$, and tag data $\mathbf{T}$. We use CAVs to discover whether the CF model has learned an implicit representation of a soft attribute corresponding to the tag. If so, that representation can be used to support example critiquing, elicitation or navigation (Sec. 5).

Critically, *we do not use the tag data* when training the CF model, akin to work that builds attribute models on top of embeddings for the cold-start problem [14, 36], and in contrast to models that jointly train attribute models [27, 44]. Our hypothesis is that *if a tag is useful for understanding user preferences across a broad swath of the population, the CF model will have learned a representation of the corresponding soft attribute.* The converse is that if no such representation (or CAV) is uncovered, this soft attribute is of limited use for users expressing their preferences. Our approach has a number of advantages: (1) the RS model can be developed/trained/used without a pre-commitment to a specific attribute vocabulary—new attributes can be added as needed; (2) RS model capacity is focused on the core task of preference prediction and recommendation; and (3) our method can be used to assess the relevance and importance of specific attributes for preference elicitation or critiquing.

3.1 Linear Attributes

We adapt CAVs to test whether a CF model has learned a representation of a soft attribute corresponding to a tag. We first illustrate our approach by testing whether the *embedding space itself* contains a *linear* representation of the tag's underlying attribute (i.e., linear w.r.t. item embedding features ϕ_I). We generalize this linear model (whose weaknesses we detail below) in Sec. 3.2. Given CF model Φ, each $i \in \mathcal{I}$ is represented by its embedding $\phi_I(i) \in X$. For any user u, the items $\phi_I(i), i \in T_{u,g}$ to which she has applied tag g are treated as positive examples of the underlying concept (say, violent movies), while $\phi_I(i), i \in T_{u,\overline{g}}$ are negatives.[4]

Our first model assumes each u uses g in roughly the same way, with positive instances given by the multi-set $\cup_{\mathcal{U}}\{\phi_I(i) : i \in T_{u,g}\}$, and negatives by $\cup_{\mathcal{U}}\{\phi_I(i) : i \in T_{u,\overline{g}}\}$. Since positive tag examples are often sparse, we use *negative sampling* to manage class imbalance [29]. Let D_g be the induced "global" (cross-user) data set. We train a logistic regressor ϕ_g, where $P(g(i); \phi_g) = \sigma(\phi_g^\top \phi_I(i))$ is the predicted probability that i "satisfies" g, using (regularized) logistic loss (and labels $y \in \{+1, -1\}$):

$$\mathcal{L}(\phi_g; D_g) = \sum_{(i,y)\in D_g} \log(1 + e^{-y\phi_g^\top \phi_I(i)}) + \frac{\lambda}{2}\phi_g^\top \phi_g. \qquad (1)$$

If two users disagree on the application of tag g to some item, this global classifier treats it as label noise. An alternative explanation for such a discrepancy is that they agree on the "direction" of g, but disagree on the "degree" to which item i exhibits g's underlying soft attribute. For example, two users may agree on which movie, for *any* pair of movies, is more violent, but have different *thresholds* or tolerances when applying the tag (i.e., disagree on "how violent is violent"). Our second model accounts for this by treating each u as generating *pairwise comparisons* $D_u = \{\phi_I(i) \succ_g \phi_I(j) : i \in T_{u,g}, j \in T_{u,\overline{g}}\}$, drawn from an underlying ranking. We use a *per-user* pairwise ranking loss to generate a *regressor* over X specifying the *degree* to which items exhibit the soft attribute:[5]

$$\mathcal{L}(\phi_g; D_{\mathcal{U}}) = \sum_{u} \sum_{\substack{i\in T_{u,g}\\ j\in T_{u,\overline{g}}}} \log(1 + e^{-\phi_g^\top(\phi_I(j)-\phi_I(i))}) + \frac{\lambda}{2}\phi_g^\top \phi_g. \qquad (2)$$

[3] While not our aim in this work, we can also use CAVs to test the sensitivity of rating predictions to the presence of this soft attribute: by analogy with testing the sensitivity of a 'zebra' classification to the presence of stripes, we can test the sensitivity of a user's item rating to the item's (degree of) violence. In the CF setting however, this sensitivity will differ across the user population.

[4] These negatives are "implicit," but plausible, since u has otherwise tagged these items.

[5] This logistic pairwise loss is as in RankNet [10]; see also LambdaRank [11] below.

The regressor ϕ_g obtained serves as our CAV. Notice ϕ_g is linear in learned item embedding features ϕ_I.

Given a CAV ϕ_g, the degree to which an item i satisfies the induced attribute is given by the score $\phi_g(i) = \phi_g^\top \phi_I(i)$. The *quality* $Q(\phi_g; D)$ of a CAV on dataset D is the fraction of the tag applications that it orders "correctly," i.e., if $i \in T_{u,g}, j \in T_{u,\bar{g}}$, then $\phi_g(i) \geq \phi_g(j)$. We can use quality Q, training/test error, or other performance metrics (see Sec. 3.3) as a measure of CAV "usefulness."

3.2 Nonlinear Attributes

A limitation of the linear approach is that if a CAV for tag g is linear in the latent embedding space $X \subseteq \mathbb{R}^d$, every user's utility for g is also linear in X. For example, if the CAV for 'violent' is linear, any user's preference would be such that she prefers either maximally or minimally violent movies—she cannot prefer movies that are "somewhat violent." Real-world preferences are often nonlinear (e.g., saturating [15]) and even non-monotone (e.g., single-peaked [31]) w.r.t. many natural attributes. Such attributes are unlikely be adequately represented linearly in X. Fortunately, CAVs can also apply to nonlinear DNN representations.

We assume a two-tower/dual-encoder model and extract CAVs from hidden layers of the item DNN N_i. Following Kim et al. [21], we assume that relevant concepts, if learned, can be uncovered within a single hidden layer of the (trained) deep CF model. Given positive and negative examples as in Sec. 3.1, we use activation $\phi_{I,\ell}(i)$ of the ℓth layer of N_I as training input instead of item embedding $\Phi_I(i)$.[6] Otherwise, the regressor is trained as above.

The result is a regressor $\phi_{g,\ell}$ that can be applied to an item's representation in the intermediate "activation space" X_I^ℓ, where $\phi_{g,\ell}^\top \phi_{I,\ell}(i)$ captures the degree to which i satisfies the induced attribute. The projection of $\phi_{g,\ell}$ through the last $L - \ell$ layers of N_I generates a (nonlinear) manifold in embedding space X, offering much more flexibility to user utilities for soft attributes.

3.3 Empirical Assessment of CAV Quality

We first evaluate our approach on synthetic data, which allows control over the generative process and access to ground truth, then test it on real-world data. For linear soft attributes, we train a CF model $\Phi = (\phi_U, \phi_I)$ using *weighted alternating least squares (WALS)* [19], with the following regularized objective:

$$(\phi_U^*, \phi_I^*) \in \arg\min \sum_{u,i} c_{u,i}(\hat{r}_{u,i} - r_{u,i})^2 + \kappa(||\phi_U||^2 + ||\phi_I||^2). \quad (3)$$

Here $c_{u,i}$ is a confidence weight for the predicted rating $\hat{r}_{u,i} = \phi_U^\top(u; \theta_U)\phi_I(i; \theta_I)$, and $\kappa > 0$ is a regularization parameter. We select embeddings (ϕ_U^*, ϕ_I^*) using validation loss and use an item-oriented confidence weight $c_{u,i} \propto m - \sum_u r_{u,i}$ (i.e., lower weight for less-frequently or lower-rated items). For nonlinear attributes, we train a two-tower DNN embedding model with SGD/Adam [22]. Further details on synthetic data generation are provided in Appendix A.2). Additional details on data generation and training methods are provided the extended version of this paper [17].

Synthetic Data. To construct synthetic data, a generative model outputs both user-item ratings $\mathbf{R}$ and tag data $\mathbf{T}$ for $n = 25,000$ users and $m = 10,000$ items. Users and items are represented by $d = 25$ dimensional embedding vectors, sampled from pre-defined mixture distributions to induce correlation in the data. For linear utility, user ratings are generated by first sampling items (giving a sparse ratings matrix $\mathbf{R}$) and then their ratings (noise added to the user/item dot product). In the nonlinear case, utility is the sum of nonlinear sub-functions (one per dimension) peaked at some (random) point and dropping as the item moves away from that peak. Users are more likely to rate items with higher utility.

To generate tags, five of the 25 latent item dimensions are treated as user-interpretable or "taggable," each with a different tag. Each u has a random *propensity to tag*, influencing the probability of tagging a rated item, and is more likely to tag higher-rated items. Each tag g has a *fixed* (non-subjective) threshold τ_g: u noisily applies g to an item if it meets τ_g.

MovieLens Data We transform and filter the MovieLens20m dataset to focus on tags with sufficient usage:[7] This leaves a 164 tags for evaluation. We split rating and tag data into train and test sets such that *all examples* for any specific user-item pair are present in exactly one of these subsets. We use a roughly $(0.75, 0.25)$ train-test split of user-item-tag triple.

CAV Accuracy. We evaluate CAV accuracy using prediction quality w.r.t. user tag usage on held-out test data. The synthetic model also allows evaluation relative to the *ground-truth item representations and attribute levels of each tag*. We evaluate three training methods, binary logistic regression, RankNet [10], and LambdaRank [11]. We measure CAV accuracy using: (i) *Accur.*, the mean accuracy of the logistic model, or *quality* $Q(\phi_g; D)$ of the ranking model; and (ii) *Sprm*, the *Spearman rank correlation coefficient* between predicted and ground-truth attribute values.

Synthetic Results. Table 1 shows the performance of the CAVs on synthetic data for three settings: (i) user utility is linear; (ii) utility is nonlinear but we train linear CAVs (Lin-Emb); and (iii) utility is nonlinear and we train nonlinear CAVs (NL-Emb). Results are averaged over the five tags. CAVs predict user tagging behavior (Accur) reasonably accurately, and reliably order test items w.r.t. their ground truth attribute values (Sprm), despite the noise in the tagging process. We bold the best values in each of the three settings. The ranking methods, RankNet and LambdaRank, dominate logistic regression w.r.t. both Accur and Sprm, which suggests that accounting for variation in user tagging behavior is important (see also the next section). For nonlinear utilities, we also compare the best "linear" CAV (extractable from the output embedding) with that the best nonlinear CAV (extractable from DNN hidden layers). Nonlinear CAVs outperform their linear counterparts, showing the value of seeking nonlinear (or "distributed") attribute representations within the DNN, and the power of TCAV to interpret them.

[6] We treat ℓ as a tunable hyperparameter in our experiments. Results for non-linear CAVs are based on the best "layerwise" CAV.

[7] We transform all tags to lowercase, and filter data to include only the user-item-tags with a rating of at least 4. Tag data is very sparse: only 268 distinct tags are applied to at least 50 unique movies. We restrict CAV training to the top 250 tags in terms of unique tagged movies. Inspection shows that tags that are applied by only a few users tend to be overly-specific or overly-generic. To exclude these, we further filter the data to include only the top 250 tags w.r.t. unique users who have used the tag at least once.

We also include a baseline tag recommender *PITF (pairwise interaction tensor factorization)* [36], which uses tensor decomposition to model pairwise interactions between users, items and tags. Its tag prediction accuracy is worse than that of the CAV approaches.

MovieLens Results. We evaluate our methods on the more realistic MovieLens20M dataset [18]. Tags are user-generated descriptions of movie attributes (e.g., genres like 'sci-fi;' qualities like 'emotional,' 'atmospheric;' themes like 'zombies,' 'cyberpunk'). We generate 50-dimensional user and item embeddings (WALS if linear, two-tower DNNs if nonlinear). Positive examples for tag g are user-item pairs to which g has been applied; negatives are those tagged by that user, but not with g. Table 2 shows test accuracy for linear and nonlinear CAVs: the ranking methods outperform logistic regression, which again hints at some subjectivity (see next section). While we cannot measure Spearman correlation (since we have no ground truth ranking) nor control the form of user utility, we see that nonlinear CAVs perform slightly better than linear CAVs, suggesting that user preferences for some MovieLens tags are nonlinear in their embedding-space representation. As above, the CAV methods consistently outperform PITF.

4 IDENTIFYING SUBJECTIVE ATTRIBUTES

If users largely agree on the usage of the tags, it is reasonable to treat the semantics of a tag as a *single* soft attribute or CAV as we do above. But in many cases, different users may have different "senses" in mind when they apply a tag. For example, one user may use the term 'funny' to describe movies that are silly, involving, say, physical or slapstick humor, while another may use the same term to refer to dry, political satire. While correlated, these two *tag senses* will order movies quite differently. Such *sense subjectivity* may hinder our ability to produce an accurate CAV and understand a user's true intent. We now turn to this issue.

Subjectivity of Degree. As discussed above, *degree subjectivity* is likely to emerge quite naturally. The use of *intra-user pairwise comparisons* with a ranking loss in CAV training ensures that the induced CAVs are robust to this form of subjectivity. However, since two users may use a tag g differently if they have different thresholds for applying g, interpreting u's usage requires a *personalized semantics* that is sensitive to her threshold. Let g be a tag that is degree subjective, ϕ_g be g's CAV and $\phi_g(i)$ be the degree to which i satisfies ϕ_g. A *user-specific threshold* $\tau_g^u \in \mathbb{R}$ determines a semantics for g: g applies (typically, noisily) to i only if $\phi_g(i) \geq \tau_g^u$.[8] The (estimated) *optimal* τ_g^u minimizes the number of misclassifications:

$$\tau_g^u \in \arg\min_{\tau} |\{i \in T_{u,g} : \phi_g(i) \geq \tau\} \cup \{i \in T_{u,\bar{g}} : \phi_g(i) < \tau\}|. \quad (4)$$

That is, the threshold τ_g^u, among the continuum of minimizers, maximizes the margin between the nearest positive and negative items. Since tag usage by an individual user is typically sparse, these thresholds are likely to be noisy. But one can reduce the noise by refining u's semantics with well-chosen queries, e.g., "Do you consider item movie m to be violent?" This can be used to implement a loose binary search to approximate the threshold (possibly made robust to account for noisy responses), but we defer this to future research. If usage is correlated within user sub-populations, generalization of thresholds across users is also viable, as we discuss in the case of sense subjectivity below.

Subjectivity of Sense. We now turn to *sense subjectivity*. We can readily detect sense subjectivity and assign a *personalized semantics* for a tag g to different (groups of) users, using *distinct CAVs* for each tag sense. We assume that g has *at most* s_g distinct senses $g[1], \ldots g[s_g]$, for some small positive integer s_g, where each sense denotes a different soft attribute (we discuss their relation below). Moreover, each user adopts exactly *one* such sense of g. We propose a method to discover whether a tag has multiple senses, and to uncover suitable CAVs for each sense if so.

Intuitively, if $Q(\phi_g; D)$ is high, then CAV ϕ_g well explains usage of g among users in dataset D. If not, then model Φ is unlikely to have learned a good representation for g. This could be due to g being poorly correlated with user ratings (hence, preferences), or because g has multiple (say, s) senses. In the latter case, there should be a user-partitioning of D into subsets $D_1, \ldots D_s$ s.t. there is a CAV $\phi_{g,k}$ with high quality $Q(\phi_{g,k}; D_k)$ for each $k \leq s$. We first propose a simple scheme to find a good set of CAVs for a fixed s, then discuss determination of a suitable number of senses $s \leq s_g$.

Assume a fixed number of target senses s and a given data set D. Let $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ be a partitioning of users into s clusters, with σ_k a set of users that (presumably) adopt a common sense for g. Let D_k be the restriction of D to tag data for $u \in \sigma_k$. For a fixed Σ, we can readily generate a CAV $\phi_{g,k}$ for each data set D_k capturing the corresponding sense, and measure its quality $Q(\phi_{g,k}; D_k)$. Of course, this quality depends on whether the partitioning Σ is sensible (i.e., whether most users in each cluster use g similarly). If the quality of these CAVs is low, we can repartition users by "assigning" each u to the cluster in Σ whose CAV best explains her tag usage:

$$k_u^* = \arg\max_k |\{(i,j) : i \in T_{u,g}, j \in T_{u,\bar{g}}, \phi_{g,k}(i) \geq \phi_{g,k}(j)\}|. \quad (5)$$

This leads to a EM-like alternating optimization procedure [4] for finding a good clustering that repeatedly: (a) learns a CAV for each current (user) cluster; then (b) reconstructs the clusters by assigning each user to the CAV that best explains her tag usage. The iterative process proceeds until Σ no longer changes or quality improvements become sufficiently small. It is easy to see that the EM procedure terminates in a finite number of steps. If we assign each u by minimizing its incurred logistic/ranking loss, convergence properties of standard k-means (e.g., [6]) show that the procedure converges to a local minimum and generates s distinct CAV senses.

We can search for the appropriate number of senses—effectively a form of *model selection* [4]—by starting with an initial (single) CAV, and applying the procedure above to gradually increasing numbers of clusters $s = 2, 3, \ldots s_g$, terminating once the improvement in average quality, $\sum_k (|D_k|/|D|) Q(\phi_{g,k}; D_k)$ is negligible.

We take a top-down "disaggregative" clustering approach, since bottom-up agglomerative clustering is likely to be very noisy—the tag set of any individual user is extremely sparse, so attempts to produce a CAV for very small groups of users will generally be unreliable. It is straightforward to assign senses to new users, and to update senses as new users, items and tagging data arises.

[8] Equivalently, this can be viewed as a *personal* linear separator for u in X, but constrained to be orthogonal to the direction ϕ_g induced from the *population* labels.

	Linear		NonLin, Lin-Emb		Nonlin, NL-Emb	
	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm
Log. Regr.	0.906	0.569	0.889	0.565	0.922	0.577
RankNet	**0.968**	0.674	0.943	**0.670**	**0.978**	**0.686**
LambdaNet	0.961	**0.679**	**0.947**	0.666	0.974	0.680
PITF	0.683	0.056	0.707	0.070	N/A	N/A

Table 1: CAV Evaluation, Synthetic Data (Non-subjective)

	Lin-Emb	NL-Emb
	Accur.	Accur.
Log. Regr.	0.727	0.745
RankNet	0.803	**0.820**
LambdaNet	**0.804**	0.818
PITF	0.715	N/A

Table 2: CAV Evaluation, MovieLens

	Linear		NonLin, Lin-Emb		Nonlin, NL-Emb	
	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm
Log. Regr.	0.872	0.566	0.860	0.523	0.886	0.548
RankNet	0.960	**0.671**	**0.947**	**0.660**	**0.961**	0.680
LambdaNet	**0.962**	0.669	0.938	0.653	0.958	**0.684**
PITF	0.700	0.064	0.708	0.068	N/A	N/A

Table 3: CAV Evaluation, Synthetic Data (Degree subjectivity)

	Linear		NonLin, LinEmb		Nonlin, NLEmb		Linear		NonLin, LinEmb		Nonlin, NLEmb	
	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm	Accur.	Sprm
Log. Regr.	0.643	0.039	0.634	0.026	0.616	0.036	0.936	0.634	0.926	0.642	0.922	0.635
EM Log. Regr.	0.833	**0.478**	0.796	0.419	0.828	0.476	0.937	0.631	0.926	**0.637**	0.922	0.635
RankNet	0.615	0.042	0.606	0.035	0.603	0.022	**0.992**	**0.636**	**0.987**	0.636	**0.982**	**0.636**
EM RankNet	**0.922**	0.466	**0.915**	**0.468**	**0.908**	**0.468**	**0.992**	0.633	**0.987**	0.633	**0.982**	**0.636**
LambdaRk	0.615	0.028	0.606	0.036	0.604	0.009	**0.992**	0.634	**0.987**	0.632	**0.982**	0.634
EM LambdaRk	0.920	0.458	**0.915**	0.465	**0.908**	0.465	**0.992**	0.631	**0.987**	0.630	**0.982**	0.631
PITF	0.672	0.063	0.711	0.070	N/A	N/A	0.640	0.050	0.675	0.074	N/A	N/A

Table 4: CAV Evaluation, Synthetic Data (Sense Subjectivity): Subjective Tags (left half), Objective Tags (right half).

Synthetic Results. We again test our approach on synthetic data to exploit access to ground truth CAV semantics. The model is similar to that used in Sec. 3.3, differing only in the addition of subjectivity to user tagging behavior. To test degree subjectivity, we use five tags as above, but with each user's personal tagging threshold sampled from a mixture distribution with two components. To test sense subjectivity, we introduce a subjective tag "tag-S" with three senses, each reflecting one of three (of the five) taggable dimensions. Each user adopts one of these three senses—when applying tag-S, they assess it based on their assigned dimension. The remaining two tags are objective. We evaluate the three CAV training methods used in Sec. 3.3, applying each to a linear (WALS) model and a nonlinear two-tower model as needed. For sense subjectivity, we test our EM-like algorithm with each training method.

Table 3 summarizes performance of the CAVs under degree subjectivity, using the same methods and models as in Sec. 3.3 (results averaged over the five degree-subjective tags). In contrast to the non-subjective case in Sec 3.3, where users have the same threshold for each tag, here the per-user ranking-based methods (RankNet and LambdaRank) significantly outperform logistic regression, demonstrating the need to be sensitive to a user's degree subjectivity.

Table 4 summarizes results for sense subjectivity, showing CAV accuracy for our baseline methods both with and without our EM-based approach for distinguishing senses. The left side of the table shows results for the sense-subjective *tag-S*, demonstrating that EM can dramatically improve CAV accuracy by disentangling the three distinct senses. This shows that treating a subjective concept as objective can be problematic. Note also that the ranking methods perform better than logistic regression. The right side shows accuracy on the two *objective* tags: the EM and non-EM methods perform almost identically, indicating that we are unlikely to identify spurious senses. The use of nonlinear CAVs offers little improvement over linear CAVs with ranking methods, though they perform better when trained using logistic regression. The performance of the PITF baseline is worse than that of the CAV approaches in both the degree and sense subjectivity experiments.

MovieLens Results. We also evaluate our subjective CAV methods on MovieLens20M. To assess degree subjectivity, we select 13 tags deemed to be degree subjective and compare the accuracy of different CAV methods for each (Table 5). Since MovieLens data has no ground truth w.r.t. possible subjectivity, Spearman rank correlation cannot be measured. Generally the ranking methods outperform logistic regression, suggesting that real users exhibit some variation in their thresholds (degree subjectivity), with some tags (e.g., *sci-fi*) having much higher agreement and CAV-predictability than others (e.g., *funny*). We also see differences in the improvement offered by nonlinear CAVs vs. linear CAVs across the tags: those with larger improvements (e.g., *dark comedy*, *dystopia*) suggest that user utility may be nonlinear in the degree of that attribute (so extreme degrees may not be preferred); while those where nonlinear CAVs perform no better, or even worse (e.g., *sci-fi*, *action*, *funny*), may be most preferred at their maximum or minimum degree.

For sense subjectivity, we construct two types of *artificial tags* from MovieLens data. First are *objective tags* capturing four *genres* (comedy, horror, fantasy, romance). For random user-item pairs, we add a genre tag if the item's meta-data lists that genre. These tags are objective—their presence does not depend on a user's tag interpretation. We also add a synthetic tag *odd year*—was a movie's release year even/odd—to 50% of user-item pairs. This (presumably)

	sci-fi	atmo-spheric	surreal	twist ending	action	funny	classic	dark comedy	quirky	psych-ology	dystopia	stylized	thought-provoking
Log. Regr.	0.831	0.721	0.739	0.705	0.823	0.689	0.818	0.714	0.725	0.715	0.737	0.753	0.764
RankNet	0.905	0.793	0.811	0.817	**0.899**	**0.775**	0.877	0.822	**0.840**	0.812	0.850	0.866	0.834
LambdaNet	**0.906**	0.784	0.788	0.869	0.876	0.605	0.838	0.830	0.821	0.788	0.867	0.809	0.839
NL. Log. Regr.	0.831	0.725	0.742	0.711	0.812	0.681	0.821	0.705	0.751	0.704	0.807	0.811	0.764
NL. RankNet	0.893	0.838	**0.827**	0.854	0.890	0.754	**0.888**	0.868	0.833	**0.837**	0.882	0.856	0.844
NL. LambdaNet	0.891	**0.843**	0.807	**0.875**	0.880	0.744	0.865	**0.891**	0.825	0.796	**0.921**	**0.898**	**0.856**

Table 5: CAV Accuracy Evaluation, 13 Possible Subjective Concepts in MovieLens

preference-irrelevant attribute serves as a baseline for which no good CAV should be discoverable.

The second artificial tag type are *sense-conflated tags*, constructed by coalescing several related "ground" tags into a single "meta-tag," then replacing each ground tag with that meta-tag. Each ground tag in the group can be viewed as a subjective sense of the meta-tag. We test our ability to "disentangle" the different senses of the meta-tag relative to the ground truth. We introduce four meta-tags: *monsters* (*zombies*, *ghosts* and *vampires*), *funny movie* (*parody*, *satire*, *dark humor*), *intrigue* (*corruption*, *conspiracy*, *politics*) and *relationship* (*family*, *friendship*, *love story*). For each user and meta-tag, we choose *exactly one* ground tag from the group as that user's *designated sense* and add the meta-tag to each user-item-tag triple that uses the ground tag (e.g., some users have all their *friendship* tags replaced with *relationship*, others have *love story* replaced).

Table 6 shows the accuracy of our trained CAVs for these two artificial tag types. For the four "objective" *genre* tags, the ranking-based methods outperform logistic regression; but using EM with ranking provides little incremental benefit. This implies that *genres* exhibit no sense subjectivity, as expected. The 'horror' and 'fantasy' tags are easiest to learn, suggesting they are more "objective" and "linear." The artificial *odd year* tag has no decent CAV, corroborating our hypothesis that CAVs are useful for identifying *preference-related* attributes/tags. By contrast, results on *sense-conflated* tags clearly demonstrate that our EM approach can disentangle distinct personal senses of each meta-tag (with each baseline method), greatly improving tag prediction accuracy.

5 USING CAVS FOR EXAMPLE CRITIQUING

While preference elicitation in RSs often uses attributes [5, 33, 41], an important question is the extent to which such methods can be adapted to handle soft attributes [34]. We do not consider this question in its full depth, but examine how the CAV semantics for tags can be used for *example critiquing* [13]. In lieu of live experiments, we adopt a stylized but plausible *user response model* in which a user's critiques are driven by her underlying utility and her personal tag/attribute semantics.[9] We run both synthetic experiments for which we have ground truth user utility and semantics, and a MovieLens setup, for which we propose a novel method for generating utilities and responses.

We assume a predefined list of critiquable tags and an interactive RS that supports user critiques (see Appendix A.3 and the extended paper [17] for more details). At each iteration with user u (maximum T steps), the RS presents a slate S of k items. u can *accept* one of the items (thence, the session terminates). Otherwise, u can *critique* S using a tag g and a desired direction 'more' or 'less' (e.g., "more funny" or "less violent"). The RS then updates its *user representation* and generates the next recommended slate. User interactions assume a *user response model* in which u has a (personal) ground truth (i) utility function over items (u can assess an item's utility if recommended) and (ii) semantics for attributes (u can assess an item's attributes). User u also has a rough estimate of the max/min levels any tag/attribute can attain in the item corpus. When presented with slate S, u accepts an item i if its utility is sufficiently large. Otherwise, u critiques using the *most salient tag* g w.r.t. utility improvement of S, i.e., $g = \text{argmax}_g\, \delta_u^T w_g$, where $\delta_u = (\phi_I(i_u^*) - \frac{1}{|S|}\sum_{i\in S}\phi_I(i)) \odot \phi_U(u)$ is the utility difference vector between u's estimated ideal item i_u^* and her average utility vector over the k items in S, and w_g is u's interpretation of g. The RS strategy for updating user embeddings is described in Appendix A.3. If tag g is sense-subjective, the critique is interpreted by the RS w.r.t. its estimate of u's sense given past usage.

We first analyze CAVs using the synthetic models above, where a user's critiques are generated with her ground truth utility and tag semantics. By contrast, the RS uses its *estimated user embedding* and the *tag's CAV* to interpret a critique (not the ground truth). We set $k = 10$ and $T = 25$, and assess how recommendation quality improves with the number of critiques by measuring *user max utility* of the top-k slate, i.e., $UMU(S) = \max_{i\in S} U(i)$, and *user average utility* $UAU(S) = \text{avg}_{i\in S} U(i)$, where $U(i)$ is u's *true* utility for i.

Fig. 1 presents interactive critiquing results in three experiments with different synthetic data sets: the first has no subjectivity and linear utility; the second, no subjectivity and nonlinear utility; the third, degree subjectivity and nonlinear utility. In all experiments, user utility or *UMU* improves with more critiquing steps, eventually converging to a steady-state value. These results corroborate our hypothesis that, since CAVs represent soft attributes well in embedding space, they can be used to effectively update an RS's beliefs about user preferences as the user critiques recommended items, which in turn improves recommendation quality. Furthermore, recall that CAVs trained with logistic regression generally have lower accuracy than those trained with RankNet or LambdaRank. While our CAV algorithms are not optimized to support critiquing, more accurate CAVs learned using ranking methods give rise to better interpretations of user critiques (this observation is reflected by both faster improvement and greater steady-state values for *UMU*). This is most likely due to the fact that more accurate ranking-based CAVs better capture a user's intended semantics during critiquing. Similarly, the improved accuracy of nonlinear CAVs when utility is nonlinear is manifest in the improved critiquing performance (in both the objective and degree-subjectivity tests). Again, this performance improvement is likely due to the better CAV representation uncovered from the intermediate layers of the DNN.

[9] Synthetic user models are often used to evaluate RSs [20, 48].

	OddYr	Comedy	Horror	Fantasy	Romance	Monsters	FunnyMovie	Intrigue	Relationship
LogRegr LinEmb	0.519	0.521	0.770	0.685	0.693	0.671	0.658	0.669	0.662
EM LogRegr LinEmb	0.532	**0.685**	0.759	0.744	0.704	0.831	0.769	0.712	0.730
RankNet LinEmb	0.505	0.620	0.790	0.778	0.730	0.718	0.705	0.660	0.634
EM RankNet LinEmb	**0.593**	0.676	0.833	**0.824**	0.749	**0.892**	**0.874**	0.834	0.840
LambdaRk LinEmb	0.533	0.609	0.809	0.779	0.716	0.719	0.718	0.661	0.623
EM LambdaRk LinEmb	0.582	0.670	**0.838**	0.819	**0.762**	0.883	0.870	**0.836**	**0.847**

Table 6: CAV Accuracy Evaluation, Artificial MovieLens Tags (5 objective, 4 sense-conflated)

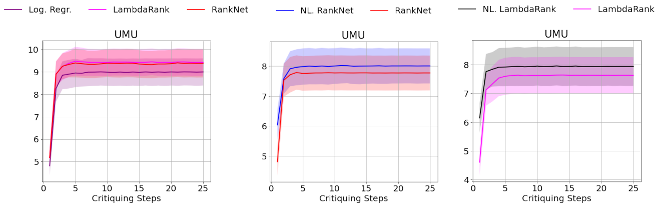

Figure 1: Results of Interactive Critique with Synthetic Data. Left two: No Subj., Lin. Utility, All Methods; Middle two: No Subj., Nonlin. Utility, RankNet; Right two: Degree Subj., Nonlin. Utility, LambdaRank

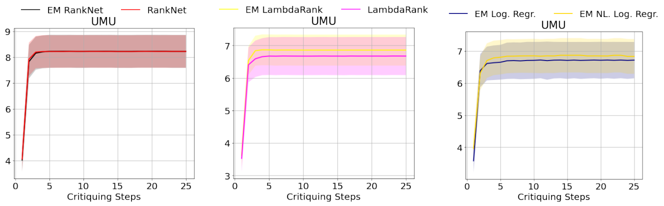

Figure 2: Critiquing (Sense-subjective, Synthetic). Left: Lin. Util., RankNet. Middle: Nonlin. Util., LambdaRk, Lin-Emb; Right: Nonlin. Util., LogRegr., NL-Emb.

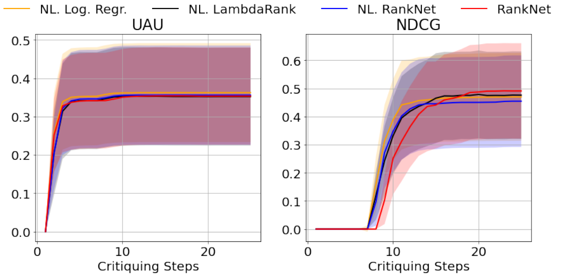

Figure 3: Interactive Critiquing with MovieLens Data

Fig. 2 shows interactive critiquing results using synthetic data with *sense subjectivity* with (i) linear utility, (ii) nonlinear utility with linear CAVs, and (iii) nonlinear CAVs. In all three settings, user utility or *UMU* improves with more critiquing steps, eventually converging to a steady-state value. (Similar results with *UAU* can be found in the extended paper [17].) Again, these results corroborate our hypotheses regarding the ability of CAVs to improve recommendation quality. While only three of 25 utility-relevant dimensions reflect "conflated" senses of a single tag, with nonlinear utility, EM-LambdaRank outperforms LambdaRank without EM. This suggests that disentangling sense subjectivity is important for critiquing. EM Logistic Regression can also be improved with nonlinear CAVs.

To evaluate critiquing with MovieLens data, we propose a novel method for hypothesizing "ground-truth" user utility. We first train a CF model with all users and items, then train (non-subjective) CAVs for 164 tags. We then construct a small set of *test users*, each of whom has rated at least 50 movies. We use the learned embedding $\phi_U(u)$ for each test user u as if it were their *ground truth utility* (since u has rated a large number of movies, we expect this to be reasonably stable and accurate). We then run the interactive critiquing RS by *forgetting* each test user (their ratings and tags) and treat them as a "cold start" user, who is given an generic prior embedding. (We use the average of all learned user embeddings as a prior.) This u then generates critiques of the RS slates using $\phi_U(u)$ as their true utility. Since we have no ground truth tag semantics, each u treats the RS's *learned CAV* as her semantics (admittedly giving the RS some advantage when interpreting critiques). Otherwise, the user response model is exactly as in the synthetic case. We evaluate as in the synthetic case, but user utility improvements are *estimates* using the learned embedding $\phi_U(u)$. Because of this we also assess some additional metrics (see below).

Fig. 3 shows critiquing results with the MovieLens data, reporting $UAU(S)$ and normalized discounted cumulative gain (NDCG) [39] of slates S generated during critiquing. We compare results using four sets of CAVs, trained with: RankNet; nonlinear RankNet; nonlinear LambdaRank; and nonlinear logistic regression. While all methods perform similarly w.r.t. *UAU*, nonlinear LambdaRank outperforms the others w.r.t. NDCG. This again provides evidence that (i) capturing user-critiquing behavior with CAVs can improve recommendation quality, and (ii) the performance of critiquing-based RSs improves with CAV quality.

6 CONCLUSIONS

We have presented a novel methodology for discovering the semantics of soft attribute/tag usage in RSs using CAVs. Its benefits include: (i) using a CF representation to identify attributes of most relevance to the recommendation task; (ii) distinguishing objective and subjective tag usage; (iii) identifying personalized, user-specific semantics for subjective attributes; and (iv) using this semantics to support critiquing with soft attributes. Future directions include additional study with real user critiques; developing real-world data sets with ground truth utility and personal semantics; and interactive elicitation of a user's semantics (e.g., using *pairwise tag-comparison queries* like "which of these two books is more thought-provoking").

REFERENCES

[1] Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. 2021. On Interpretation and Measurement of Soft Attributes for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. to appear..

[2] G. Bang, G. Barash, R. Bea, J. Cali, M. Castillo-Effen, X. Chen, N. Chhaya, R. Cummings, R. Dhoopar, S. Dumanci, H. Espinoza, E. Farchi, F. Fioretto, R. Fuentetaja, C. Geib, O. E. Gundersen, J. Hernández-Orallo, X. Huang, K. Jaidka, S. Keren, S. Kim, M. Galley, X. Liu, T. Lu, Z. Ma, R. Mallah, J. McDermid, M. Michalowski, R. Mirsky, S. Ó hÉigeartaigh, D. Ramachandran, J. Segovia-Aguas, O. Shehory, A. Shaban-Nejad, V. Shwartz, S. Srivastava, K. Talamadupula, J. Tang, P. Van Hentenryck, D. Zhang, and J. Zhang. 2020. The Association for the Advancement of Artificial Intelligence 2020 Workshop Program. In *AI Magazine*. 100–114.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM-18)*. Marina Del Rey, CA, 46–54.

[4] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

[5] Edwin V. Bonilla, Shengbo Guo, and Scott Sanner. 2010. Gaussian Process Preference Elicitation. In *Advances in Neural Information Processing Systems 23 (NIPS-10)*. Vancouver, 262–270.

[6] Léon Bottou and Yoshua Bengio. 1994. Convergence Properties of the K-Means Algorithms. In *Advances in Neural Information Processing Systems 7 (NIPS-94)*. 585–592.

[7] Craig Boutilier, Kevin Regan, and Paolo Viappiani. 2009. Online Feature Elicitation in Interactive Optimization. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML-09)*. Montreal, 73–80.

[8] Craig Boutilier, Kevin Regan, and Paolo Viappiani. 2010. Simultaneous Elicitation of Preference Features and Utility. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence (AAAI-10)*. Atlanta, 1160–1167.

[9] Craig Boutilier, Richard S. Zemel, and Benjamin Marlin. 2003. Active Collaborative Filtering. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI-03)*. Acapulco, 98–106.

[10] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank using Gradient Descent. In *Proceedings of the Twenty-second International Conference on Machine Learning (ICML-05)*. 89–96.

[11] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. *Learning* 11, 23–581 (2010), 81.

[12] Robin Burke. 2002. Interactive Critiquing for Catalog Navigation in E-Commerce. *Artificial Intelligence Review* 18, 3–4 (2002), 245–267.

[13] Li Chen and Pearl Pu. 2012. Critiquing-based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 125–150.

[14] Deborah Cohen, Michal Aharon, Yair Koren, Oren Somekh, and Raz Nissim. 2017. Expediting Exploration by Attribute-to-feature Mapping for Cold-start Recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys17)*. Como, Italy, 184–192.

[15] Simon French. 1986. *Decision Theory*. Halsted Press, New York.

[16] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *2010 IEEE International Conference on Data Mining (ICDM-10)*. 176–185.

[17] Christina Göpfert, Yinlam Chow, Chih-wei Hsu, Ivan Vendrov, Tyler Lu, Deepak Ramachandran, and Craig Boutilier. 2022. Discovering Personalized Semantics for Soft Attributes in Recommender Systems using Concept Activation Vectors. (2022). arXiv:2202.02830.

[18] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2016), 19:1–19:19.

[19] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. 263–272.

[20] Guangda Huzhang, Zhen-Jia Pang, Yongqing Gao, Yawen Liu, Weijie Shen, Wen-Ji Zhou, Qianying Lin, Qing Da, An-Xiang Zeng, Han Yu, Yang Yu, and Zhi-Hua Zhou. 2021. AliExpress Learning-To-Rank: Maximizing online model performance without going online. *IEEE Transactions on Knowledge and Data Engineering* (2021). https://doi.org/10.1109/TKDE.2021.3098898

[21] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the Thirty-fifth International Conference on Machine Learning (ICML-18)*. Stockholm, 2668–2677.

[22] Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the Third International Conference on Learning Representations (ICLR-15)*. San Diego, CA.

[23] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. *Commun. ACM* 40, 3 (1997), 77–87.

[24] Jonathan Koren, Yi Zhang, and Xue Liu. 2008. Personalized Interactive Faceted Search. In *Proceedings of the 17th International Conference on World Wide Web (WWW-08)*. Beijing, 477–486.

[25] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys09)*. 61–68.

[26] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012*. 173–182.

[27] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference 2020*. 2535–2541.

[28] Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *12th International Conference on Data Mining (ICDM-12)*. 1020–1025.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*. 3111–3119.

[30] Martin Mladenov, Chih wei Hsu, Vihan Jain, Eugene Ie, Christopher Colby, Nicolas Mayoraz, Hubert Pham, Dustin Tran, Ivan Vendrov, and Craig Boutilier. 2020. Demonstrating Principled Uncertainty Modeling for Recommender Ecosystems with RecSim NG. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. 591–593.

[31] Hervé Moulin. 1980. On Strategy-proofness and Single Peakedness. *Public Choice* 35, 4 (1980), 437–455.

[32] Daniel N. Osherson and Edward E. Smith. 1981. On the Adequacy of Prototype Theory as a Theory of Concepts. In *Cognition*.

[33] Pearl Pu and Li Chen. 2008. User-involved Preference Elicitation for Product Search and Recommender Systems. *AI Magazine* 29, 4 (2008), 93–103.

[34] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *Proceedings of the Annual SIGDial Meeting on Discourse and Dialogue*.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence (UAI-09)*. Montreal, 452–461.

[36] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM-10)*. 81–90.

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems 20 (NIPS-07)*. Vancouver, 1257–1264.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*. 3319–3328.

[39] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. 2009. Learning to Rank by Optimizing NDCG Measure.. In *NIPS*, Vol. 22. 1883–1891.

[40] Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. 2020. Gradient-based optimization for Bayesian preference elicitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10292–10301.

[41] Paolo Viappiani and Craig Boutilier. 2010. Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets. In *Advances in Neural Information Processing Systems 23 (NIPS)*. Vancouver, 2352–2360.

[42] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. 2013. A Survey of Faceted Search. *Journal of Web Engineering* 12, 1&2 (2013), 041–064.

[43] Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Exploring the Value of Personalized Word Embeddings. In *Proceedingsof the 28th International Conference on Computational Linguistics*.

[44] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 137–145.

[45] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. In *Proceedings of the Web Conference (WWW-20)*. Taipei, 441–447.

[46] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, and E. Chi. 2019. Sampling-bias-corrected Neural Modeling for Large Corpus Item Recommendations. In *Proceedings of the Thirteenth ACM Conference on Recommender Systems (RecSys19)*. Copenhagen, 269–277.

[47] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive Collaborative Filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM-13)*. 1411–1420.

[48] Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. 2020. Counterfactual Prediction for Bundle Treatment. In *Advances in Neural Information Processing Systems 33 (NeurIPS-20)*.

A APPENDIX

We recap key terms and definitions and provide a graphical overview of how we construct and use CAVs in App. A.1. We provide a sketch of our synthetic data generation process in App. A.2, and offer further detail on our example critiquing set up in App. A.3.

A.1 An Overview of CAV Usage in RSs

We first recap several key concepts used in our work.

- *Rating*: measure $r_{u,i}$ of user u's preference for item i.
- *User/item embedding*: vector representations of users $\phi_U(u)$ and items $\phi_I(i)$ learned using, say, collaborative filtering on ratings data. The estimate of u's rating for i is $\hat{r}_{i,u} = \phi_U(u)^\top \phi_I(i)$. If ϕ_I is represented by a DNN, we denote the activations for item i at the ℓ-th layer by $\phi_{I,\ell}(i)$.
- *Tags*: set of terms $\mathcal{T}$ propositionally applied by users to describe items. Each tag corresponds an attribute or *concept*.
- *Concept activation vector (CAV)*: the CAV ϕ_g for a tag g is a vector in embedding or activation space that represents a direction in which items "possess more of" the concept represented by g. CAVs can be learned using classification or learning-to-rank methods on tag data.
- *Subjectivity*: we distinguish three types of tags: (1) *objective* tags, where users agree on whether (or the degree to which) an item satisfies the attribute underlying the tag; (2) *degree subjective* tags, where users agree on the degree, but may disagree on whether the (boolean) attribute/tag applies; and (3) *sense subjective* tags, where (groups of) users may disagree on which items possess the attribute.

Fig. 4 offers a graphical depiction our use of CAVs for RSs.

A.2 Synthetic Data Generation with RecSim

We use a stylized, but structurally realistic generative model to produce synthetic ratings and tag data for some of our experiments. This provides us with a "ground truth" against we can test (i) the quality of our learned CAV representations of soft attributes and (ii) the effectiveness of our elicitation methods at using soft, subjective attributes to improve recommendations.

The generative user-response model is implemented using RecSim NG [30]. We first describe the process for generating ratings and tags for "non-subjective" tags, where users have linear utility for the corresponding soft attributes. We then describe mild modifications of this core model to allow for (degree and sense) subjective tags and nonlinear attribute utility.

Some details are omitted for space reasons. We refer to the extended version of the paper for a complete specification [17].

Non-subjective, linear utility model. The generative process proceeds in stages: we first generate items (with latent and soft attribute values); then users (with utility functions); then user-item ratings; and finally user-item tags. The model reflects realistic characteristics such as item and user "clustering," popularity bias, not-missing-at-random ratings, the sparsity of ratings, the relative sparsity of tags compared to ratings, etc.

Each item i is characterized by an *attribute vector* $\mathbf{v}(i) \in [0,1]^D$, where $D = L + S$: L dimensions correspond to latent item features and S to soft attributes. For a soft dimension $L < s \le L + S$, $v^s(i)$ captures the degree to which i exhibits attribute s. We sample m items from a mixture of K D-dimensional Gaussian distributions (truncated on $[0,1]^D$) $\mathcal{N}(\mu_k, \sigma_k)$, $k \le K$, with mean $\mu_k \in [0,1]^D$ and (diagonal) covariance σ_k. We set $D = 25, K = 100$. Each item i has a ranodm popularity bias $b_i \in [0,1]$ (see below).

Each user u has a *utility vector* $\mathbf{w}(u) \in [0,1]^D$ over items. We sample n users from a K-mixture-of-Gaussian distribution similar to that for items. Different mixture weights ensure that users and items are distributed in different parts of the latent "topic space." We generate user-item ratings as follows: (i) For each u, we draw Num_u samples from a Zipf distribution with power parameter $a = 1.05$ to reflect the natural power law over the number of ratings provided by users. (ii) To generate the candidate items to be rated by each user u, we generate a set of $Rated_u$ items by sampling them without replacement from the overall set of items via a multinomial logit (or softmax) choice model, where the probability associated with each item i is proportional to $e^{\tau \cdot (\mathbf{w}_u \mathbf{v}_i + b_i)}$. (iii) Rating r_{ui} for $i \in Rated_u$ is generated as follows. Let $s(u,i) := \mathbf{w}_u \mathbf{v}_i + \varepsilon$ be u's score of i, where ε is a small, zero-mean random noise. We then discretize all scores of u into 5 equally sized sub-intervals and assign a 1 to 5 rating to each item accordingly.

For each soft attribute s we assume a unique tag g_s that users can apply when referring to that attribute. Let $s(g)$ the corresponding soft attribute (so $g = g_{s(g)}$). We generate user-item tags as follows. (i) For each u, PT_u, the probability of tagging an item, is drawn from a mixture of (a) a Dirac at 0 with weight $0 < x < 1$; and (b) a uniform over $[p_-, p_+]$ with weight $1 - x$. This reflects that many users never use tags, and among those who do, some users tag more frequently than others. (ii) We generate the set $Tagged_u$ of items tagged by u s.t. each $i \in Rated_u$ is tagged with (independent) probability PT_u. (iii) For every (non-subjective) tag g, a user-independent threshold $\tau_g = 0.5$ indicates the degree to which an item must possess attribute $s(g)$ to be tagged with g by a user. (iv) For every $i \in Tagged_u$ and g, indicator $t_{u,i,g} = 1$ (i.e., u applied tag g to item i) if $v^{s(g)}(i) \ge \tau_g + \varepsilon$.

Subjective model. The generative model above is modified slightly to handle subjective attributes. For degree subjectivity, we generate user-dependent tag-application thresholds τ_g^u for each user-tag pair. To allow for some "commonality" across user sub-populations, we draw these thresholds from mixture distributions with a small number of components and small variance.

For sense subjectivity, we maintain S_{obj} soft attributes of the form above—which we now call *objective*. Each $s \in S_{obj}$ corresponds to one item dimension and to a specific *objective (in sense)* tag g_s. In addition, we have S_{subj} *subjective* soft attributes, partitioned into *tag groups*, $S^1, \ldots, S^J$ satisfying the following conditions: (a) $S^i \cap S^j = \emptyset$ for $i \ne j$; (b) $\cup_{j \le J} S^j = S_{subj}$; and (c) $|S^j| > 1$ for all $j \le J$. Each tag group S^j is associated with a single tag g^j, with each $s \in S^j$ reflecting a different *sense* for g^j.

For each tag group S^j, each user u is randomly assigned to exactly one such sense $s(u,j) \in S^j$. This has two implications. First, when user u considers applying tag g^j to an item, it is evaluated according to soft attribute $s(u,j)$. This means that u uses that specific sense when applying that tag. Second, the utility vector $\mathbf{w}(u)$ of user u is such that its s^{th} component is zero for each $s \in S^j$ except for $s(u,j)$. This implies that u assesses her utility for an item using only her designated attribute (or sense) from each of the tag groups.

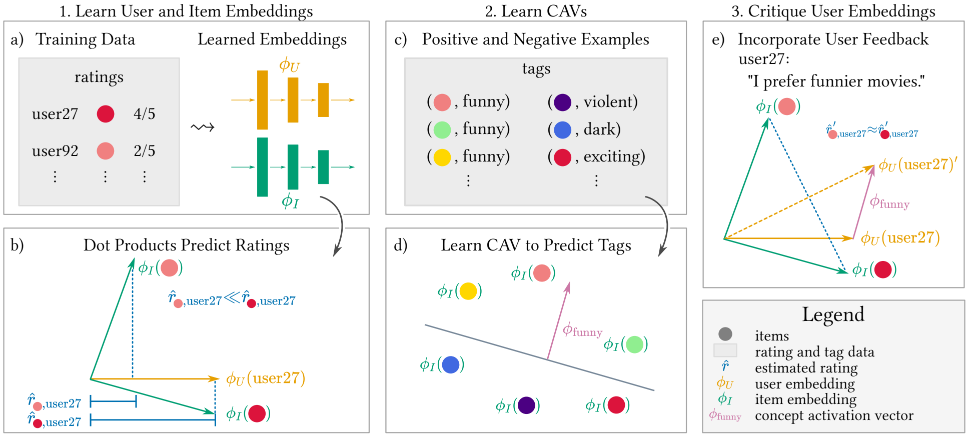

Figure 4: An overview of the CF, CAV learning and critiquing setup used in our work. a) Learn user ϕ_U and item embeddings ϕ_I from ratings data. b) User-item affinity $\hat{r}_{i,u}$ is predicted using dot products of user-item embeddings. c) To learn a CAV for the concept 'funny,' gather positive and negative examples, e.g., from tag data, and use them to d) learn a CAV in item-embedding space. (We explore several methods for CAV training. e) In one use case, we use the CAV to update a user's embedding given her item-attribute feedback. Figure inspired by [21].

Nonlinear utility model. We consider single-peaked utility functions, and for simplicity apply them only to non-subjective attributes. Extending it to the subjective case is straightforward. Assume user utility is *additive-independent* across attributes, i.e., the utility for an item is the sum of "local utilities" for each attribute (the dot-product model satisfies this trivially). A user u's utility function is *single-peaked* w.r.t. s if u has an ideal point $p_{u,a} \in [0,1]$ such that u's local utility for attribute $l_{u,a}(x)$ is maximized at $p_{u,a}$ and decreases monotonically as x moves away from $p_{u,a}$. The functional form of a single-peaked utility can be arbitrary with the simplest form being piecewise linear. Here we use $l_{u,a}(x) = p_{u,a} - |x - p_{u,a}|$, where $x = \mathbf{w}_{u,a}\mathbf{v}_{i,a} \in [0,1]$. We sample $p_{u,a}$ from a uniform distribution $U(L_a, 1)$, where L_a is a user-specific per-attribute parameter. In the special case when $L_a = 1$, utility is linear. In our experiments, we set L_a to 0.3 for sense-subjective tags and L_a to 0.5 for the rest.

A.3 Additional Critiquing Details

We fill in a few additional details of the critiquing experiments in Sec. 5. In the synthetic data experiment, we construct each user's *estimated ideal item* using the knowledge of her ground-truth utility function, which is either linear or single-peaked linear in each (latent or soft-attribute) dimension. We note that a user's (actual or estimated) "ideal" item may not actually exist in the item corpus $\mathcal{I}$. Insisting that the user's ideal item exist is unrealistic, since it requires a dense item space. Moreover, the user generally does not know the identify of the actual best item in $\mathcal{I}$, since this would assume too great a state of knowledge for most users. Instead, we assume each u has a rough estimate of the maximum and minimum levels any tag/attribute can attain in the item corpus and uses this to drive her critiques. Note that when user utility is linear, the ideal item must occur at the boundary of item space, so the estimates inform her estimated ideal. In the MovieLens experiment, we use the ratings data to derive an estimated ground-truth utility for each *test user* (who must have rated sufficiently many items as described in the main text).

During the critiquing process, the RS updates its user embedding based on the user's response.[10] We assume item embeddings $\phi_I(i)$ are fixed, and use a *simple heuristic RS strategy* for incorporating critiques.[11] Given a user embedding $\phi_U(u)$, the RS scores all items i in the corpus w.r.t. utility $r_{i,u} = \phi_I(i)^T\phi_U(u)$, and presents the slate S of the k top scoring items.

Suppose at step $t > 0$ the user critiques S with a specific tag g (and a specific direction, *more* or *less*). The RS updates the user embedding in response to this critique using a simple heuristic update function: $\phi_U(u) \leftarrow \phi_U(u) + \text{Sgn} \cdot \alpha_t(g) \cdot \phi_g$. Here Sgn $\in \{+1, -1\}$ indicates the direction of the move (+1 more, −1 less), and $\alpha_t(g)$ is a step size that controls the scale of the move of the user embedding in the direction of tag g's CAV. For each g, we decay the step size $\alpha_t(g)$ with each critique using g, $\alpha_t(g) = \alpha_0(g)/(1+t)$, where α_0 is g's initial step size. This ensures that the updating process converges to a stable point (and does not cycle or repeat slates of items) given the coarse control mechanism offered to the user. If the tag g is sense-subjective, the critique is interpreted relative to the RS's estimate of u's sense (or user cluster) based on past usage.

In our experiments, $\alpha_0(g)$ is constant (α_0) across all tags, and we treat it as a tunable hyper-parameter selected to optimize user utility metrics such as *UMU* and *UAU* utilities.[12] The critiquing results we report are based on the set of hyper-parameters optimized using a validation set.

[10] We use the average of all learned user embeddings as the RS's prior.

[11] We emphasize that the RS strategy used and method for incorporating critiques are fairly generic and are not intended to reflect the state-of-the-art, since our goal is to measure the ability to exploit learned CAVs. More elaborate strategies for updating user embedding are possible, including the use of Bayesian updates relative to a prior over the user embedding [40]. Here instead we adopt a simple heuristic, based on [27], to focus attention on the CAV semantics itself.

[12] Ultimately, this heuristic adjustment should be tuned to the specifics of a real-world user response models.