Preference Learning and Ranking

Johannes Fürnkranz, TU Darmstadt
Eyke Hüllermeier, Paderborn University

Synonyms

learning from preferences, comparison training, constraint classification

Definition

Preference learning refers to the task of learning to predict (contextualized) preferences on a collection of alternatives, which are often represented in the form of an order relation, on the basis of observed or revealed preference information. Supervision in preference learning is typically weak, in the sense that only partial information about sought structures or indirect information about an underlying value function are provided; a common example is feedback in the form of pairwise comparisons between alternatives. Especially important in preference learning are ranking problems, in which preferences are represented in terms of total or partial order relations. Such problems can be approached in two fundamentally different ways, either by learning binary preferences on pairs of alternatives or by inducing an underlying (latent) value function on single alternatives.

Motivation and Background

Preference information plays a key role in automated decision making and appears in various guises in Artificial Intelligence (AI) research [8]. In particular, the formal modeling of preferences can be considered an essential aspect of autonomous agent design. Yet, in spite of the existence of formalisms for representing preferences in a compact way, such as CP-networks [1], modeling preferences by hand is a difficult task. This is an important motivation for preference *learning*, which is meant to support and partly automatize the design of preference models. Roughly speaking, preference learning is concerned with the automated acquisition of preference models from observed or revealed preference information, that is, data from which (possibly uncertain) preference representations can be deduced in a direct or indirect way.

Computerized methods for revealing the preferences of individuals (users) are useful not only in AI, but also in many other fields showing a tendency

for *personalization* of products and services, such as computational advertising, e-commerce, and information retrieval, where such techniques are also known as learning to rank [19]. Correspondingly, a number of methods and tools have been proposed with the goal of leveraging the manifold information that users provide about their preferences, either explicitly via ratings, written reviews, etc., or implicitly via their behavior (shopping decisions, websites visited, and so on). Typical examples include recommender systems and collaborative filtering, which can be viewed as special cases of preference learning. A first attempt at setting a common framework for this emerging subfield of machine learning was made by Fürnkranz and Hüllermeier [11].

Ranking is one of the key tasks in the realm of preference learning. One can distinguish between two important types of ranking problems, namely, learning from object and learning from label preferences. A ranking is a special type of preference structure, namely a *strict total order*, that is, a binary relation $\succ$ on a set $\mathcal{A}$ of alternatives that is total, irreflexive, and transitive. In agreement with our preference semantics, $a \succ b$ suggests that alternative a is preferred to alternative b. However, in a wider sense, the term "preference" can simply be interpreted as any kind of order relation. For example, $a \succ b$ can also mean that a is an algorithm that outperforms b on a certain problem, or that a is a student finishing her studies before another student b.

Structure of the Learning System

An important difference between object and label ranking concerns the formal representation of the preference context and the alternatives to be ordered. In object ranking, the alternatives themselves are characterized by properties, typically in terms of a feature vector (attribute-value representation). Thus, the learner has the possibility to generalize via properties of the alternatives, whence a ranking model can be applied to arbitrary sets of such alternatives. In label ranking, the alternatives to be ranked are labels as in classification learning, i.e., mere identifiers without associated properties. Instead, the ranking context is characterized in terms of an instance from a given instance space, and the task of the model is to rank alternatives depending on properties of the context. Thus, the context may change (as opposed to object ranking, where it is implicitly fixed) but the objects to be ranked remain the same. Stated differently, object ranking is the problem to rank varying sets of objects under invariant preferences, whereas label ranking is the problem to rank an invariant set of objects under varying preferences.

Both problems can be approached in two principal ways, either by learning a *value function* that induces the sought ranking by *evaluating individual alternatives*, or by comparing pairs of alternatives, that is, learning a *binary preference relation*. Note that the first approach implicitly assumes an underlying total order relation, since numerical (or at least totally ordered) utility scores enforce the comparability of alternatives. The second approach is more general in this regard, as it also allows for partial order relations. On the other hand, this ap-

proach may lead to additional complications, since a set of *hypothetical* binary preferences induced from empirical data may exhibit inconsistencies in the form of preferential cycles.

Learning from Object Preferences

The most frequently studied problem in learning from preferences is to induce a *ranking function* $r(\cdot)$ that is able to order any (finite) subset $\mathcal{O}$ of an underlying (possibly infinite) class $\mathcal{X}$ of objects. That is, $r(\cdot)$ assumes as input a subset $\mathcal{O} \subseteq \mathcal{X}$ of objects and returns as output a permutation τ of $\{1, \ldots, |\mathcal{O}|\}$. The interpretation of this permutation is that, for objects $x_i, x_j \in \mathcal{O}$, the former is preferred to the latter whenever $\tau(i) < \tau(j)$. The objects themselves are typically characterized by a finite set of features as in conventional attribute-value learning. The training data consists of a set of exemplary pairwise preferences $x \succ x'$ with $x, x' \in \mathcal{X}$. A survey of object ranking approaches is given by Kamishima et al. [18].

Note that, in order to evaluate the predictive performance of a ranking algorithm, an accuracy measure (or loss function) is needed that compares a predicted ranking with a given reference ranking. To this end, one can refer, for example, to statistical measures of rank correlation. Expected or empirical loss minimization is a difficult problem for measures of that kind, especially because they are not (instance-wise) decomposable.

Many learning to rank problems may be viewed as object ranking problems. For example, Joachims [17] studies a scenario where the training information could be provided implicitly by the user who clicks on some of the links in a query result and not on others. This information can be turned into binary preferences by assuming a preference of the selected pages over those nearby pages that are not clicked on. Applications in information retrieval typically suggest loss functions that put more emphasis on the top and less on the bottom of a ranking; for this purpose, specific measures have been proposed, such as the (normalized) discounted cumulative gain [19].

Learning from Label Preferences

In label ranking, preferences are contextualized by elements x of an instance space $\mathcal{X}$, and the goal is to learn a ranking function $\mathcal{X} \longrightarrow \mathcal{S}_m$ for a fixed $m \geq 2$. Thus, for any instance $x \in \mathcal{X}$ (e.g., a person), a prediction in the form of an associated ranking $\succ_x$ of a finite set $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$ of labels or alternatives is sought, where $\lambda_i \succ_x \lambda_j$ means that instance x prefers λ_i to λ_j. Again, the quality of a prediction of that kind is typically captured in terms of a rank correlation measure (or an associated loss function). The training information consists of a set of instances for which (partial) knowledge about the associated preference relation is available. More precisely, each training instance x is associated with a subset of all pairwise preferences. Thus, despite the assumption of an underlying ("true") target ranking, the training data is not expected to provide full information about such rankings (and may even

contain inconsistencies, such as pairwise preferences that are conflicting due to observation errors).

The above formulation essentially follows Fürnkranz and Hüllermeier [10], though similar formalizations have been proposed independently by several authors; for an overview, see the survey papers by Vembu and Gärtner [23] and Zhou et al. [24]. Label ranking contributes to the general trend of extending machine learning methods to complex and structured output spaces [22]. Moreover, label ranking can be viewed as a generalization of several standard learning problems. In particular, the following well-known problems are special cases of learning label preferences: (i) Classification, where a single class label λ is assigned to each instance x; this is equivalent to the set of preferences $\{\lambda \succ_x \lambda_j \,|\, \lambda_j \in \mathcal{L} \setminus \{\lambda\}\}$. (ii) Multi-label classification, where each training example x is associated with a subset $L \subseteq \mathcal{L}$ of possible labels. This is equivalent to the set of preferences $\{\lambda_i \succ_x \lambda_j \,|\, \lambda_i \in L, \lambda_j \in \mathcal{L} \setminus L\}$. In each of the former scenarios, the sought prediction can be obtained by post-processing the output of a ranking model $f : \mathcal{X} \longrightarrow \mathcal{S}_m$ in a suitable way. For example, in multi-class classification, where only a single label is requested, it suffices to project a label ranking to the top-ranked label.

Applications of this general framework can be found in various fields, for example in marketing research; here, one might be interested in discovering dependencies between properties of clients and their preferences for products. Another application scenario is meta-learning, where the task is to rank learning algorithms according to their suitability for a new dataset, based on the characteristics of this dataset [20]. Moreover, every preference statement in the well-known CP-nets approach [1], a qualitative graphical representation that reflects conditional dependence and independence of preferences under a *ceteris paribus* interpretation, formally corresponds to a label ranking function that orders the values of a certain attribute depending on the values of the parents of this attribute (predecessors in the graph representation).

Other Settings

A nunber of variants of the above ranking problems have been proposed and studied in the literature. For example, a setting referred to as *instance ranking* is very similar to object ranking. However, instead of relative (pairwise) comparisons, training data consists of absolute ratings of alternatives; typically these ratings are taken from an ordinal scale, such as 1 to 5 stars. Moreover, a predicted ranking is not compared with another (ground-truth) ranking but with the multi-partition induced by the rating of the alternatives [12].

Attempts have also been made at combining object and label ranking, that is, to exploit feature representations of both the preference context and the alternatives to be ranked. One approach is to combine both pieces of information by means of a *joint feature map* $\phi : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathcal{Z}$ and to learn a value function $f : \mathcal{Z} \longrightarrow \mathbb{R}$; here, $\mathcal{Y}$ is a parametric or structured space of alternatives and $\mathcal{Z} \subseteq \mathbb{R}^d$ a joint feature space [22, 20].

Learning Utility Functions

Evaluating alternatives in terms of a value or utility function is a very natural way of representing preferences, which has a long tradition in economics and decision theory [9]. In the object preferences scenario, such a function is a mapping $f : \mathcal{X} \longrightarrow \mathbb{R}$ that assigns a utility degree $f(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$. In the label preferences scenario, a utility function $f_i : \mathcal{X} \longrightarrow \mathcal{U}$ is needed for every label λ_i, $i = 1, \ldots, m$. Here, $f_i(x)$ is the utility assigned to alternative λ_i in the context x. To obtain a ranking for x, the alternatives are ordered according to their utility scores, i.e., a ranking $\succ_x$ is derived such that $\lambda_i \succ_x \lambda_j$ implies $f_i(x) \geq f_j(x)$.

If the training data offers the utility scores directly, preference learning essentially reduces to a standard regression or an ordinal regression problem, depending on the underlying utility scale. This information can rarely be assumed, however. Instead, usually only constraints derived from comparative preference information of the form "this alternative should have a higher utility score than that alternative" are given. Thus, the challenge for the learner is to find a value function that is as much as possible in agreement with a set of such constraints.

For object ranking approaches, this idea has first been formalized by Tesauro [21] under the name *comparison training*. He proposed a symmetric neural-network architecture that can be trained with representations of two states and a training signal that indicates which of the two states is preferable. The elegance of this approach comes from the property that one can replace the two symmetric components of the network with a single network, which can subsequently provide a real-valued evaluation of single states. Similar ideas have also been investigated for training other types of classifiers, in particular support vector machines. We already mentioned Joachims [17] who analyzed "click-through data" in order to rank documents retrieved by a search engine according to their relevance. Earlier, Herbrich et al. [14] proposed an algorithm for training SVMs from pairwise preference relations between objects.

For the case of label ranking, a corresponding method for learning the functions $f_i(\cdot)$, $i = 1, \ldots, m$, from training data has been proposed in the framework of *constraint classification*, which allows for reducing a label ranking to a single binary classification problem [13]. The learning method proposed in this work constructs two training examples, a positive and a negative one, for each given preference $\lambda_i \succ_x \lambda_j$, where the original N-dimensional training example (feature vector) x is mapped into an $(m \times N)$-dimensional space. In this space, the learner finds a linear model (hyperplane) f that separates the positive from the negative examples. Finally, the model f is "split" into m linear value functions $f_1, \ldots, f_m$, one for each label.

Learning Preference Relations

An alternative to learning latent utility functions consists of learning binary preference relations, which essentially amounts to reducing preference learning

to binary classification. For object ranking, the pairwise approach has been pursued in [7]. The authors propose to solve object ranking problems by learning a binary preference predicate $Q(x, x')$, which predicts whether x is preferred to x' or vice versa. A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these (possibly conflicting) predictions.

For label ranking, the pairwise approach has been introduced in [16] as a natural extension of *pairwise classification*, a well-known class binarization technique. The idea is to train a separate model (base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L}$, $1 \leq i < j \leq m$; thus, a total number of $m(m-1)/2$ models is needed. For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$. Moreover, $y = 1$ if $i < j$ and $y = 0$ otherwise. Thus, $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$. This mapping can be realized by any binary classifier. Instead of a $\{0, 1\}$-valued classifier, one can of course also employ a scoring classifier. For example, the output of a probabilistic classifier would be a number in the unit interval $[0, 1]$ that can be interpreted as a probability of the preference $\lambda_a \succ_x \lambda_b$.

At classification time, a query $x_0 \in \mathcal{X}$ is submitted to the complete ensemble of binary learners. Thus, a collection of predicted pairwise preference degrees $\mathcal{M}_{i,j}(x)$, $1 \leq i, j \leq m$, is obtained. The problem, then, is to turn these pairwise preferences into a ranking of the label set $\mathcal{L}$. To this end, different ranking procedures can be used. The simplest approach is to extend the (weighted) voting procedure that is often applied in pairwise classification: For each label λ_i, a score

$$S_i = \sum_{1 \leq j \neq i \leq m} \mathcal{M}_{i,j}(x_0)$$

is derived (where $\mathcal{M}_{i,j}(x_0) = 1 - \mathcal{M}_{j,i}(x_0)$ for $i > j$), and then the labels are ordered according to these scores. Despite its simplicity, this ranking procedure has several appealing properties. Apart from its computational efficiency, it turned out to be relatively robust in practice and, moreover, it possesses some provable optimality properties in the case where Spearman's rank correlation is used as an underlying accuracy measure. Roughly speaking, if the binary learners are unbiased probabilistic classifiers, the simple "ranking by weighted voting" procedure yields a label ranking that maximizes the expected Spearman rank correlation [15]. Finally, it is worth mentioning that, by changing the ranking procedure, the pairwise approach can also be adjusted to accuracy measures other than Spearman's rank correlation.

Other Approaches

Referring to the type of training data and the loss function to be minimized on this data, learning value functions and learning preference relations are sometimes called the "pointwise" and "pairwise" approach to preference learning, respectively. This is distinguished from the "listwise" approach, in which a loss is defined on a predicted ranking directly. This can be done, for example, on

the basis of probabilistic models of ranking data, such as the Plackett-Luce model. The idea, then, is to learn the parameters of a probabilistic model using statistical methods such as maximum likelihood estimation (or, equivalently, minimizing logarithmic loss). Methods of this kind have been proposed both for object ranking [3] and label ranking [5].

Yet another alternative is to resort to the idea of local estimation techniques as prominently represented, for example, by the nearest neighbor estimation principle: Considering the rankings observed in similar situations as representative, a ranking for the current situation is estimated on the basis of these neighbor-rankings, namely, by finding a suitable consensus among them; essentially, this is a problem of rank aggregation [4].

Future Directions

As already said, preference learning is an emerging branch of machine learning and still developing quite dynamically. In particular, new settings or variants of existing frameworks will certainly be proposed and studied in the future. As for ranking problems, for example, an obvious idea and reasonable extension is to go beyond strict total order relations and instead allow for *incomparability* or *indifference* between alternatives, and for representing uncertainty about predicted relations [6]. Another interesting direction is to combine preference learning with online learning, i.e., to predict preferences in an online setting. First steps in the direction of online preference learning have recently been made with a preference-based variant of the multi-armed bandit problem [2].

See Also

Class Binarization, Classification, Meta-Learning, Multi-armed bandit, Multi-label Classification, Online Learning, Rank Correlation, Regression

Recommended Reading

[1] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of AI Research*, 21:135–191, 2004.

[2] R. Busa-Fekete and E. Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Proc. ALT, 25th International Conference on Algorithmic Learning Theory*, pages 18–39, Bled, Slovenia, 2014. Springer-Verlag.

[3] Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proc. ICML, 24th International Conference on Machine learning*, pages 129–136, 2007.

[4] W. Cheng, J. Hühn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proc. ICML–2009, 26th International Conference on Machine Learning*, pages 161–168, Montreal, Canada, 2009.

[5] W. Cheng, K. Dembczynski, and E. Hüllermeier. Label ranking based on the Plackett-Luce model. In *Proc. ICML–2010, International Conference on Machine Learning*, pages 215–222, Haifa, Israel, 2010.

[6] W. Cheng, E. Hüllermeier, W. Waegeman, and V. Welker. Label ranking with partial abstention based on thresholded probabilistic models. In *Proc. NIPS–2012, 26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, US, 2012.

[7] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10(1):243–270, 1999.

[8] C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. *Artificial Intelligence*, 175(7–8):1037–1052, 2011.

[9] P.C. Fishburn. *Utility-Theory for Decision Making*. Wiley, 1969.

[10] J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010.

[11] J. Fürnkranz and E. Hüllermeier. Preference learning: An introduction. In *Preference Learning*, pages 1–18. Springer-Verlag, 2010.

[12] J. Fürnkranz, E. Hüllermeier, and S. Vanderlooy. Binary decomposition methods for multipartite ranking. In *Proc. ECML/PKDD–2009, European Conference on Machine Learning and Knowledge Discovery in Databases*, Bled, Slovenia, 2009.

[13] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: a new approach to multiclass classification. In *Proc. 13th Int. Conf. on Algorithmic Learning Theory*, pages 365–379, Lübeck, Germany, 2002. Springer-Verlag.

[14] R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Supervised learning of preference relations. In *Proceedings des Fachgruppentreffens Maschinelles Lernen (FGML-98)*, pages 43–47, 1998.

[15] E. Hüllermeier and J. Fürnkranz. On predictive accuracy and risk minimization in pairwise label ranking. *Journal of Computer and System Sciences*, 76(1):49–62, 2010.

[16] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD–02, 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM Press, 2002.

[18] T. Kamishima, H. Kazawa, and S. Akaho. A survey and empirical comparison of object ranking methods. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*, pages 181–202. Springer-Verlag, 2010.

[19] T.Y. Liu. *Learning to Rank for Information Retrieval*. Springer-Verlag, 2011.

[20] D. Schäfer and E. Hüllermeier. Dyad ranking using a bilinear Plackett-Luce model. In *Proc. ECML/PKDD–2015, European Conference on Machine Learning and Knowledge Discovery in Databases*, Porto, Portugal, 2015.

[21] G. Tesauro. Connectionist learning of expert preferences by comparison training. In *Advances in Neural Information Processing Systems 1 (NIPS-88)*, pages 99–106. Morgan Kaufmann, 1989.

[22] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[23] S. Vembu and T. Gärtner. Label ranking: a survey. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*. Springer-Verlag, 2010.

[24] Y. Zhou, Y. Lui, J. Yang, X. He, and L. Liu. A taxonomy of label ranking algorithms. *Journal of Computers*, 9(3), 2014.

Rank Correlation

Definition

Rank correlation measures the correspondence between two rankings τ and τ' of a set of m objects. Various proposals for such measures have been made, especially in the field of statistics. Two of the best-known measures are Spearman's Rank Correlation and Kendall's tau:
Spearman's Rank Correlation [26] calculates the sum of squared rank distances and is normalized such that it evaluates to -1 for reversed and to $+1$ for identical rankings. Formally, it is defined as follows:

$$(\tau, \tau') \mapsto 1 - \frac{6 \sum_{i=1}^{m} (\tau(i) - \tau'(i))^2}{m(m^2 - 1)} \tag{1}$$

Kendall's tau [25] is the number of pairwise rank inversions between τ and τ', again normalized to the range $[-1, +1]$:

$$(\tau, \tau') \mapsto 1 - \frac{4 \left| \{(i,j) \mid i < j,\, \tau(i) < \tau(j) \wedge \tau'(i) > \tau'(j)\} \right|}{m(m-1)} \tag{2}$$

Spearman's rank correlation and Kendall's tau give equal weight to all ranking positions, which is not desirable for all applications. For example, ranking problems in information retrieval are often evaluated with the (normalized) discounted cumulative gain (NDCG), which assigns more weight to the lower ranking positions (cf. learning to rank).

See Also

Learning to Rank, Preference Learning, ROC Analysis

Recommended Reading

[25] Kendall, M. (1938). A New Measure of Rank Correlation. *Biometrika 30*(1):81–89.

[26] Spearman C (1904). The proof and measurement of association between two things. *American Journal of Psychology 15*:2–101.