Preference Learning

Luis Muñoz González

Machine Learning Group
University Carlos III of Madrid
Spain

December 12, 2012

Index

Some starting references on Preference Learning

- **Preference Learning site:** http://www.preference-learning.org
- **Research Groups:** Univ. degli Studi di Padova (Fabio Aiolli), Universiteit Gent (Bernard De Baets), TU Darmstadt (Johannes Furnkranz), Philipps-Universitat Marburg (Eyke Hullermeier), Cornell University (Thorsten Joachims), Microsoft Research Asia (Tie-Yan Liu), MIT (Cynthia Rudin).
- **Recent Workshops:** Preference Learning: Problems and Applications in Artificial Intelligence (PL-12) at ECAI-12, Choice Models and Preference Learning (CMPL-11) at NIPS-11, Preference Learning (PL-10) at ECML/PKDD 2010, Yahoo! Learning to Rank Challenge Workshop at ICML-10.
- **Special issues:**
  - Eyke Hullermeier, Johannes Furnkranz, Special Issue on Preference Learning and Ranking, Machine Learning.
  - Tie-Yan Liu, Thorsten Joachims, Hang Li and Chengxiang Zhai, Special Issue on Learning to Rank for Information Retrieval, Information Retrieval 13(3), 2010.

Definition

- **From Wikipedia:** "Preference learning is a subfield in machine learning in which the goal is to learn a predictive preference model from observed preference information. In the view of supervised learning, preference learning trains on a set of items which have preferences toward labels or other items and predicts the preferences for all items".
- **Tasks:** we can distinguish three types of ranking problems in preference learning
  - Label ranking
  - Instance ranking
  - Object ranking

Label Ranking

- **Given:**
  - A set of training instances $\{\mathbf{x}_\ell | \ell = 1, 2, ..., n\} \subseteq \mathcal{X}$
  - A set of labels $\mathcal{Y} = \{y_i | i = 1, 2, ..., k\}$
  - For each training instance $\mathbf{x}_\ell$, a pairwise preferences of the form $y_i \succ_{\mathbf{x}_\ell} y_j$
- **Find** a ranking function that maps any $\mathbf{x} \in \mathcal{X}$ to a ranking $\succ_{\mathbf{x}}$ of $\mathcal{Y}$.
- **Performance measures:**
  - Ranking error (e.g., based on rank correlation measures) comparing predicted ranking with target ranking.
  - Position error comparing predicted ranking with a target label.

Label Ranking

Some learning problems may be formulated in terms of label preferences:

- **Classification:** A single class label y_i is assigned to each sample $\mathbf{x}_\ell$. This implicitly defines the set of preferences $\{y_i \succ_{\mathbf{x}_\ell} y_j | 1 \leq j \neq i \leq k\}$.
- **Multi-label classification:** Each training example $\mathbf{x}_\ell$ is associated with a subset $P_\ell \subseteq \mathcal{Y}$ of possible labels. This implicitly defines the set of preferences $\{y_i \succ_{\mathbf{x}_\ell} y_j | y_i \in P_\ell, y_j \in \mathcal{Y} \backslash P_\ell\}$.

Instance Ranking

- **Given:**
  - A set of training instances $\{\mathbf{x}_\ell | \ell = 1, 2, ..., n\} \subseteq \mathcal{X}$
  - A set of labels $\mathcal{Y} = \{y_i | i = 1, 2, ..., k\}$ endowed with an order $y_1 < y_2 < ... < y_k$
  - For each training instance $\mathbf{x}_\ell$ an associated label y_ℓ
- **Find** a ranking function that allows to order a new set of instances $\{\mathbf{x}_j\}_{j=1}^{t}$ according to their (unknown) preference degrees.
- **Performance measures:**
  - The area under the ROC curve (AUC) in the dichotomous case ($k = 2$)
  - Generalizations such as the C-index in the polychotomous case ($k > 2$).

Object Ranking

- **Given:**
  - A (potentially infinite) reference set of objects $\mathcal{Z}$
  - A finite set of pairwise preferences $\mathbf{x}_i \succ \mathbf{x}_j$, $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{Z} \times \mathcal{Z}$
- **Find** a ranking function $f(\cdot)$ that assumes as input a set of objects and returns a permutation (ranking) of this set.
- **Performance measures:**
  - Ranking error (e.g., based on rank correlation measures) comparing the predicted ranking with the target ranking.
  - Top-K measures comparing the top-positions of the rankings.
  - Retrieval measures such as precision, recall, NDCG.

Learning Utility functions

- Utility function assigns an abstract degree of utility to each alternative under consideration. Depending on the underlying utility scale the problem can be considered as:
  - Regression: if the utility scale is numerical.
  - Classification: if the utility scale is ordinal.
- For **instance and object preferences** scenario, utility function is a mapping $f : \mathcal{X} \rightarrow \mathcal{R}$ that assigns a utility degree $f(\mathbf{x})$ to each instance (object) $\mathbf{x}$ and, hence, induces a complete order on $\mathcal{X}$.
- For **label ranking** a utility function $f_i : \mathcal{X} \rightarrow \mathcal{R}$ is needed for each of the labels y_i. Thus, $f_i(\mathbf{x})$ is the utility assigned to alternative y_i by instance $\mathbf{x}$.

Learning Utility functions

- For **instance ranking**, the scores of the utility function are given in the data.
- For **label** and **object ranking** the training data does not provide direct information about the utility function. The challenge is to find a function which is as much as possible in agreement with the constraints given in the data.

Learning Preference Relations

- The objective is to learn a binary preference relation that compares pairs of alternatives (objects or labels).
- The training of the model is simpler, because comparative training information can be used directly (instead of translating into constraints on the utility function).
- However, predictions may become more difficult, since a binary preference relation learned from data is not necessarily consistent in the sense of being transitive (so, it does not define a ranking in a unique way).
- The difficulty of this optimization problem depends on the concrete criterion used. Many natural objectives can lead to NP hard problems.

Learning Preference Relations

Some efficient techniques:

- Simple voting (Borda count procedure).
- Weak (instead of strict) linear orders (e.g., *bucket order*).
- Ranking by pairwise comparison (for label ranking problems). Deriving pairwise preferences via weighted voting (generalized Borda counting).

Other techniques

- **Model-based preference learning:** Ranking functions are learned from specific model assumptions about the structure of the preference relations.
- **Local aggregation of preferences:** Rankings are estimated locally (similar to K-NN): considering the rankings observed in similar situations as representative, a ranking for the current situation is estimated on the basis of these neighboured rankings, typicallly using an averaging-like aggregation operator.

Applications of Preference Learning

There are two particularly interesting problems in preference learning (among others):

- **Learning to rank search results:** given a query $\mathbf{q}$ and a set of documents $\mathcal{D}$, find a ranking of the documents in $\mathcal{D}$ that corresponds to their relevance with respect to $\mathbf{q}$. Preference relations are obtained from user feedback on past rankings.
- **Recommender systems:** users may provide a preference degree explicitly giving some feedback (as a rating) or implicitly (e.g., buying the product).

Preference Learning with Gaussian Processes

Reference: Chu, W. and Ghahramani, Z., *Preference Learning with Gaussian Processes*, Procs. of ICML 2005. Bonn, Germany.

- Bayesian model proposed for instance and label ranking.
- A likelihood function is proposed to capture preference relations (i.e., learning the utility function).
- The method is also applicable to multiclass problems.

Instance ranking GP model

Consider:

- A set of n instances $\mathbf{x}_i \in \mathcal{R}^d$ denoted as $\mathcal{X} = \{\mathbf{x}_i : i = 1, ..., n\}$
- A set of m observed pairwise preference relations of the instances, denoted as $\mathcal{D} = \{v_k \succ u_k : k = 1, ..., m\}$ where $v_k \in \mathcal{X}$ and $u_k \in \mathcal{X}$.
- An unobservable latent (utility) function is assumed, where $f(\mathbf{x}_i)$ is the value associated with each training sample $\mathbf{x}_i$.
- Following a Bayesian treatment, we are interested in computing the posterior probability for the utility function. So that,

$$p(\mathbf{f}|\mathcal{X}, \mathcal{D}) = \frac{p(\mathbf{f}|\mathcal{X})p(\mathcal{D}|\mathbf{f}, \mathcal{X})}{p(\mathcal{D}|\mathbf{X})}$$

The likelihood

For ideal noise-free cases, we could assume this likelihood:

$$p_{\text{ideal}}(v_k \succ u_k | f(v_k), f(u_k)) = \begin{cases} 1 & \text{if } f(v_k) \geq f(u_k) \\ 0 & \text{otherwise} \end{cases}$$

To assume some tolerance to noise in the inputs of preference relations, we could assume that the latent (utility) function is contaminated with Gaussian noise with zero mean and σ^2 variance. With these considerations, the likelihood becomes

$$p(v_k \succ u_k | f(v_k), f(u_k)) = \Phi\left(\frac{f(v_k) - f(u_k)}{\sqrt{2}\sigma}\right)$$

Posterior approximation

The posterior probability can be written as

$$p(\mathbf{f}|\mathcal{X},\mathcal{D}) = \frac{p(\mathbf{f}|\mathcal{X})}{p(\mathcal{D}|\mathbf{X})}\prod_{k=1}^{m} p(v_k \succ u_k|f(v_k), f(u_k))$$

- The posterior and the evidence are not Gaussian, so that, inference is not tractable analytically.
- Laplace approximation is proposed to approximate the posterior with a Gaussian distribution at the maximum a posteriori (MAP) estimate, i.e.,

$$\mathbf{f}_{\text{MAP}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathcal{X},\mathcal{D})$$

which is equivalent to the minimizer of the following functional

$$\mathcal{S}(\mathbf{f}) = -\sum_{k=1}^{m} \ln\Phi\left(\frac{f(v_k) - f(u_k)}{\sqrt{2}\sigma}\right) + \frac{1}{2}\mathbf{f}^T\Sigma^{-1}\mathbf{f}$$

Posterior approximation

The Newton-Raphson formula can be used to find the solution for simple cases. as $\frac{\partial \mathcal{S}(\mathbf{f})}{\partial \mathbf{f}}|_{\mathbf{f}_{\text{MAP}}} = 0$

$$\mathbf{f}_{MAP} = \Sigma \boldsymbol{\beta}$$

with

$$\boldsymbol{\beta} = \frac{\partial \sum_{k=1}^{m} \ln \Phi \left(\frac{f(v_k) - f(u_k)}{\sqrt{2}\sigma} \right)}{\partial \mathbf{f}}$$

This lead to a Gaussian posterior approximation given by

$$q(\mathbf{f}|\mathcal{X}, \mathcal{D}) = \mathcal{N}(\mathbf{f}|\mathbf{f}_{MAP}, (\Sigma^{-1} + \Lambda_{MAP}))$$

Posterior approximation

where Λ is a $n \times n$ matrix whose ij-th entry is

$$\Lambda_{ij} = \frac{\partial \sum_{k=1}^{m} -\ln \Phi \left(\frac{f(v_k) - f(u_k)}{\sqrt{2}\sigma} \right)}{\partial f(\mathbf{x}_i) f(\mathbf{x}_j)}$$

and Λ_{MAP} is the matrix Λ at the MAP estimate.

Predictive model

Consider a test pair (r, s) on which the preference relation is unknown and $\mathbf{f}_t = [f(r), f(s)]^T$. Then,

$$p(\mathbf{f}_t|\mathcal{X}, \mathcal{D}) = \int p(\mathbf{f}_t|\mathbf{f})q(\mathbf{f}|\mathcal{X}, \mathcal{D})d\mathbf{f} = \mathcal{N}(\mathbf{f}_t|\mu_t, \Sigma_t)$$

with

$$\mu_t = [\mu_r \mu_s]^T = \mathbf{k}_t^T \Sigma^{-1} \mathbf{f}_{\mathrm{MAP}} = \mathbf{k}_t^T \beta$$

$$\Sigma_t = \begin{bmatrix} \Sigma_{rr} & \Sigma_{rs} \\ \Sigma_{sr} & \Sigma_{ss} \end{bmatrix} = \mathbf{K}_{tt} - \mathbf{k}_t(\mathbf{I} + \Lambda_{MAP}\Sigma)^{-1}\Lambda_{MAP}\mathbf{k}_t$$

Then, the predictive preference $p(r \succ s|\mathcal{D})$ can be evaluated by the integral

$$p(r \succ s|\mathcal{D}) = \int p(r \succ s|\mathbf{f}_t, \mathcal{D})p(\mathbf{f}_t|\mathcal{D})d\mathbf{f}_t = \Phi\left(\frac{\mu_r - \mu_s}{\sigma_t}\right)$$

where $\sigma_t = 2\sigma^2 + \Sigma_{rr} + \Sigma_{ss} - \Sigma_{rs} - \Sigma_{sr}$

Questions

Thank you for your attention!!!

Questions?