Regularized Least-Squares for Learning Non-Transitive Preferences between Strategies

Tapio Pahikkala* Evgeni Tsivtsivadze* Antti Airola*
Tapio Salakoski*
*Turku Centre for Computer Science (TUCS)
Department of Information Technology, University of Turku
Joukahaisenkatu 3-5 B, FIN-20520 Turku, Finland
firstname.lastname@utu.fi

Abstract

Most of the current research in preference learning has concentrated on learning transitive relations. However, there are many interesting problems that are non-transitive. Such a learning task is, for example, the prediction of the probable winner given the strategies of two competitors. In this paper, we investigate whether there is a need to learn non-transitive preferences, and whether they can be learned efficiently. In particular, we consider cyclic preferences such as those observed in the game of rock paper and scissors.

1 Introduction

The learning of preferences (see e.g. Fürnkranz and Hüllermeier (2005)) has recently gained significant attention in the machine learning community. Preference learning can be considered as a task in which the aim is to learn a function capable of evaluating, given pair of data points, whether the first point is preferred over the second one. For example, given two competitive strategies, the aim might be to predict the probable winner. We assume that we are given a training set of pairwise preferences that are used to train a supervised learning algorithm for the prediction of the preference relations among unseen data points.

1.1 Non-Transitive Preferences

The typical setting for preference learning deals with transitive preferences. By a transitive preference, we mean that $A > B$ and $B > C$ imply $A > C$, where $>$ denotes the preference relation, and A, B and C are objects of interest. In the commonly used scoring setting, where each object is associated with a goodness score, all preference relations that can be derived from the scores are transitive.

In this paper, we consider the learning of non-transitive preference relations. A typical example of such a relation occurs in the game rock-paper-scissors. In the game, rock defeats scissors and scissors defeat paper, but rock loses to paper.

Is there a reason to aim to learn such relations? In the context of decision theory there has been discussion about whether non-transitivity of preferences arises simply from irrationality or errors in measurements or whether reasonable preferences can actually exhibit non-transitivity (see e.g. Fishburn (1991)). Next, we present some examples that can be considered as real-world non-transitive preference learning tasks.

Some motivation for considering non-transitive preferences can be found, for example, in recent biological findings. Kerr et al. (2002); Kirkup and Riley (2004) report that this type of phenomenon appears between bacterial populations of *Eschericia coli*: bacteria that produce a certain type of antibiotic kill bacteria that are sensitive to it, but are outcompeted by bacteria resistant to it, while sensitive bacteria outcompete resistant ones. Therefore, it makes sense to aim to predict, for two new types of bacteria, which outcompetes which. A new bacteria could be, for example, of a type that produces just a small amount of antibiotic but with a lower competitive cost.

These types of relations occur not only on the bacterial level but, for example, also in the mating strategies of certain lizard species (Sinervo and Lively, 1996). Aggressive orange males outcompete their less aggressive blue peers, but are outsmarted by males with yellow markings. Yet the yellow males lose to the more perceptive blue males.

Similar examples of non-transitive preferences can also be found in military settings. For example, weapon systems like bombers, long-range artillery, and anti-aircraft batteries again form a preference cycle. In general, when a set of competing strategies are used against each other, the interplay of the weaknesses and strengths of these strategies can result in nonlinear preference relations.

Finally, non-transitive preferences are often confronted in the domain of computer games. In Crawford (1984), building nontransitive relationships into computer games was termed as "triangularity". Nowadays, triangularity is one of the most well-known design patterns in computer game development (see e.g. Björk et al. (2003)). Preference learning methods that are able to learn this type of relationships, for example, from statistics collected in a computer game may prove to be advantageous tools in adjusting the balance of the game rules and mechanics.

1.2 Related Work

Preference learning has so far concentrated on learning a scoring function either from a scored data (see e.g. Herbrich et al. (1999); Pahikkala et al. (2007); Cortes et al. (2007)) or from a given set of pairwise preferences (see e.g. Joachims (2002)). The quality of the learned scoring function is measured according to how well it performs with respect to a given ranking measure. This is sometimes called the scoring based setting.

There have also been studies about learning and using a preference function that, when given two objects, outputs a direction or magnitude of preference between them (see e.g. Cohen et al. (1999); Ailon and Mohri (2008)). However, even though such a function can be used to represent non-transitive preferences, the aim in these studies has been to obtain a total order of the objects.

Both of the aforementioned approaches are unsuitable for learning tasks in which the aim is to preserve the non-transitivities instead of turning the problem into a linear ranking task. For example, it makes no sense to consider tasks such as the learning the preferences between the rock, paper, and scissors strategies in the ranking framework.

In this paper, we adopt a third approach in which we aim to construct learners that preserve the non-transitivities. We achieve this via training nonlinear classifiers and regressors with pairs of individual objects and the corresponding directions or magnitudes of preferences between them.

Kernel-based learning algorithms (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) have been shown to be successful in solving non-linear tasks, and hence they make a good candidate for learning non-transitive preferences. However, the computational complexity of those algorithms may be high, because the number of labeled object pairs often grows quadratically with respect to the number of objects. Fortunately, there exist efficient approximation methods which output sparse representations of the learned function, such as the regularized least-squares regression (RLS) together with the subset of regressors approach (see e.g. Rifkin et al. (2003)).

2 Learning the Preferences

We formulate the preference learning task in Section 2.1. In Section 2.2, we describe the synthetic data used to simulate a non-transitive preference learning task and in Section 2.3 the learning algorithm. Experimental results are presented in Section 2.4.

2.1 Problem Formulations

Let $\mathcal{V}$ denote the set of possible inputs. Moreover, let $X = (x_1, \ldots, x_l)^{\mathrm{T}} \in (\mathcal{V} \times \mathcal{V})^l$ be a sequence of l observed preferences between the inputs and let $Y = (y_1, \ldots, y_l) \in \mathbb{R}^l$ be their corresponding magnitudes. That is, for each $x_i = (v, v')$, where $v, v' \in \mathcal{V}$, $y_i \in \mathbb{R}$ indicates the direction and the magnitude of preference between v and v'. Clearly, X can be considered as a preference graph in which the inputs are the vertices and x_i are the edges. The nontransitivity implies that the preference graph can contain cycles.

2.2 Synthetic Data

To test the performance of the learning algorithm in a nonlinear preference learning task, we generated the following synthetic data. First, we generate 100 preference graph vertices for training and 100 for testing. The preference graph vertices are three-dimensional vectors representing players of the rock-paper-scissors game. The three attributes of the players are the probabilities that the player will choose rock, paper, or scissors, respectively. The probability $P(r \mid v)$ of the player v choosing rock is determined by $P(r \mid v) = \exp(wu)/z$, where u is a random number drawn from the uniform distribution between 0 and 1, w is a steepness parameter, and z is a normalization constant ensuring that the three probabilities sum up to one. By using the exponent function

with the parameter w it can be ensured that most of the players tend to favor one of the three choices.

We generate 1000 edges for training by randomly selecting the start and end vertices from the training vertices. Each edge represents a game of rock-paper-scissors. For both players we randomly choose either rock, paper, or scissors according to their personal probabilities. The outcome of a game is -1, 0, or 1 depending on whether the first player loses the game, the game is a tie, or the first player wins the game, respectively. We use the game outcomes as the labels of the training edges.

Similarly, we generate 1000 edges for testing from the test vertices. However, instead of using the outcome of a single simulated game as a label, we assign for each test edge the average outcome of a game played between the first and the second player, that is,

$$\begin{aligned} y \;=\; & P(p \mid v)P(r \mid v') - P(s \mid v)P(r \mid v') \\ & -P(r \mid v)P(p \mid v') + P(s \mid v)P(p \mid v') \\ & +P(r \mid v)P(s \mid v') - P(p \mid v)P(s \mid v'). \end{aligned}$$

The task is to learn to predict the average outcomes of the test edges from the training data.

2.3 Learning Method

RLS is a state of the art kernel-based machine learning method which has been shown to have comparable performance to support vector machines (Rifkin et al., 2003; Poggio and Smale, 2003). We choose the sparse version of the algorithm, also known as subset of regressors, as it allows us to scale the method up to very large training set sizes.

Let us denote $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$, and let $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$ be the hypothesis space. In order to construct an algorithm that selects a hypothesis f from $\mathcal{H}$, we have to define an appropriate cost function that measures how well the hypotheses fit to the training data. Further, we should avoid too complex hypotheses that overfit at training phase and are not able to generalize to unseen data. Following Schölkopf et al. (2001), we consider the framework of regularized kernel methods in which $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) defined by a positive definite kernel function k. The kernel functions are defined as follows. Let $\mathcal{F}$ denote the feature vector space. For any mapping

$$\Phi : \mathcal{X} \to \mathcal{F},$$

the inner product

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

of the mapped data points is called a kernel function. Using RKHS as our hypothesis space, we define the learning algorithm as

$$\mathcal{A}(S) = \operatorname*{argmin}_{f \in \mathcal{H}} J(f),$$

where

$$J(f) = c(f(X), Y) + \lambda \|f\|_k^2, \tag{1}$$

$f(X) = (f(x_1), \ldots, f(x_m))^{\mathrm{T}}$, c is a real valued cost function, and $\lambda \in \mathbb{R}_+$ is a regularization parameter controlling the tradeoff between the cost on the training set and the complexity of the hypothesis. By the generalized representer theorem (Schölkopf et al., 2001), the minimizer of (1) has the following form:

$$f(x) = \sum_{i=1}^{m} a_i k(x, x_i), \tag{2}$$

where $a_i \in \mathbb{R}$.

We now briefly present the basic sparse RLS algorithm. Let $M = \{1, \ldots, m\}$ be an index set in which the indices refer to the examples in the training set. Instead of allowing functions that can be expressed as a linear combination over the whole training set as with the basic RLS regression, we only allow functions of the following restricted type:

$$f(x) = \sum_{i \in B} a_i k(x, x_i), \tag{3}$$

where k is the kernel function, $a_i \in \mathbb{R}$ are weights, and the set indexing the basis vectors $B \subseteq M$ is selected in advance. The coefficients a_i that determine (3) are obtained by minimizing

$$\sum_{i=1}^{m} (y_i - \sum_{j \in B} a_j k(x_i, x_j))^2 + \lambda \sum_{i,j \in B} a_i a_j k(x_i, x_j)$$

where the first term is the squared loss function, the second term is the regularizer, and $\lambda \in \mathbb{R}_+$ is a regularization parameter. The minimizer is obtained by solving the corresponding system of linear equations, which can be performed in $O(l|B|^2)$ time.

We set the maximum number of basis vectors to 100 in all experiments in this study, and select the subset randomly when the training set size exceeds this number, since in Rifkin et al. (2003) it was shown that randomly relecting the basis vectors works as well as heuristic-based methods.

As the kernel function, we use the Gaussian kernel over the feature vectors of the edges, which are constructed by catenating the feature vectors of its start

	$w = 1$	$w = 10$	w=100
I	0.002	0.004	0.001
II	5e-06	0.029	0.509
III	0.666	0.912	0.938

Table 1: I: The mean squared errors made by the regression algorithm. II: The mean squared errors made by always predicting 0. III: The proportions of correctly predicted directions of preference by the regression algorithm.

and end vertices. Formally, the Gaussian kernel is defined as follows:

$$k(x, x') = e^{-\gamma(x-x')^2},$$

where $\gamma > 0$ is a bandwidth parameter.

In our experiments, we set the parameters λ and γ with grid search and cross-validation. For in depth discussion of the behavior of kernel-based learning algorithms with different combinations of these parameter values, we refer to Lippert and Rifkin (2006).

2.4 Results

We conduct experiments with three data sets generated using the values 1, 10, and 100 for the w parameter. The value $w = 1$ corresponds to the situation where all probabilities of the players are close to the uniform distribution. When using $w = 100$ the players tend to always play their favourite item, and $w = 10$ corresponds to a setting between these two extremes.

The results are presented in Table 1. We report the mean square-error made by the regression algorithm when predicting the average outcome and compare it to the approach of always predicting zero. We also report the proportions of correctly predicted directions of preference for each edge. As expected, learning the average outcomes when the probabilities of the players are close to the uniform distribution is more difficult than in case the players tend to always play their favourite item. Nevertheless, the sparse RLS regressor with Gaussian kernel is capable of capturing the nonlinear concept to be learned.

3 Conclusion

In this paper, we investigate the problem of learning non-transitive preference relations. We discuss where this type of problems appear and how they can be solved. In particular, a case study about the game of rock-paper-scissors is presented. In the study, we create synthetic data for which we apply sparse RLS with Gaussian kernel which proves to be a feasible approach for the task.

In the future, we will consider other variations of nonlinear preferences occurring in the real world learning tasks and how to efficiently solve them. For example, tasks consisting of a mixture of trainsitive and non-transitive preference relations may provide interesting research directions. Further, modern computer games have often a large set of competing strategies and players with different strengths and weaknesses from which non-transitive preference relations might emerge.

Acknowledgments

This work has been supported by Academy of Finland and Tekes, the Finnish Funding Agency for Technology and Innovation.

References

Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In Rocco Servedio and Tong Zhang, editors, *Proceedings of the 21th Annual Conference on Learning Theory*, pages 87–97, 2008.

Staffan Björk, Sus Lundgren, and Jussi Holopainen. Game design patterns. In *Digital Games Research Conference DIGRA*, 2003.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. Magnitude-preserving ranking algorithms. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning*, volume 227 of *ACM International Conference Proceeding Series*, pages 169–176. ACM Press, 2007.

Chris Crawford. *The Art of Computer Game Design*. Osborne/McGraw-Hill, Berkeley, CA, USA, 1984.

Peter C Fishburn. Nontransitive preferences in decision theory. *Journal of Risk and Uncertainty*, 4(2): 113–34, April 1991.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning. *Künstliche Intelligenz*, 19(1):60–61, 2005.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. In *Proceedings of the Ninth International Conference on Articial Neural Networks*, pages 97–102. Institute of Electrical Engineers, 1999.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

Benjamin Kerr, Margaret A. Riley, Marcus W. Feldman, and Brendan J. M. Bohannan. Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature*, 418(6894):171–174, 2002.

Benjamin C. Kirkup and Margaret A. Riley. Antibiotic-mediated antagonism leads to a bacterial game of rock-paper-scissors in vivo. *Nature*, 428(6981):412–414, 2004.

Ross Lippert and Ryan Rifkin. Asymptotics of gaussian regularized least squares. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 803–810. MIT Press, Cambridge, MA, 2006.

Tapio Pahikkala, Evgeni Tsivtsivadze, Antti Airola, Jorma Boberg, and Tapio Salakoski. Learning to rank with pairwise regularized least-squares. In Thorsten Joachims, Hang Li, Tie-Yan Liu, and ChengXiang Zhai, editors, *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, pages 27–33, 2007.

Tomaso Poggio and Steve Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50(5):537–544, 2003.

Ryan Rifkin, Gene Yeo, and Tomaso Poggio. Regularized least-squares classification. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Model and Applications*, volume 190 of *NATO Science Series III: Computer and System Sciences*, chapter 7, pages 131–154. IOS Press, 2003.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In D. Helmbold and R. Williamson, editors, *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, pages 416–426, Berlin, Germany, 2001. Springer-Verlag.

Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels*. MIT Press, Cambridge, Massachusetts, 2002.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

Barry Sinervo and Curtis M. Lively. The rock-paper-scissors game and the evolution of alternative male strategies. *Nature*, 380:240–243, 1996.