Multimedia Tools and Applications (2020) 79:35093–35107
https://doi.org/10.1007/s11042-020-09184-y

A kernel machine for hidden object-ranking problems (HORPs)

Formalization and application to Tahitian black pearls quality assessment

Gaël Mondonneix[1] · Jean Martial Mari[1] · Sébastien Chabrier[1] · Alban Gabillon[1]

Received: 8 June 2019 / Revised: 12 May 2020 / Accepted: 4 June 2020 /
Published online: 3 July 2020
© The Author(s) 2020

Abstract
Hidden Object-Ranking Problems (HORPs) are object-ranking problems stated as classification or instance-ranking problems. There exists so far no dedicated algorithm for solving them properly and HORPs are usually solved as if they were classification (multi-class or ordinal) or instance-ranking problems. In the former case, item-related ordinal information is negated and only class-related information is retained; in the latter case, item-related ordinal information is considered, but in a way that emphasizes class-related information, so that the items are not only sorted but also clustered. We propose a kernel machine that allows retaining item-related ordinal information while avoiding emphasizing class-related information. We show how this kernel machine can be implemented with standard optimization libraries provided slight modifications on the original kernel. The proposed approach is tested on Tahitian pearls quality assessment and compared with four other classical methods. It yields better results ($93.6\% \pm 3.9\%$ of correct predictions without feature selection, $94.3\% \pm 3.4\%$ with feature selection) than the best of the other tested methods ($91.3\% \pm 3.4\%$ and $92.6\% \pm 4.3\%$ without and with feature selection for the instance-ranking approach), this improvement being significant (p-value < 0.05). Moreover, this method exhibits no significant difference in the results with and without feature selection (p-value $= 0.33$), which may be a hint that its learning bias fits the problem well and can thus alleviate the data preprocessing workload.

Keywords Ordinal learning · Classification · Object-ranking · Instance-ranking · Learning bias · Kernel machine · Support vector machine · Pearls quality assessment

1 Introduction

Some learning problems happen to be often mistaken for other kinds of problems, and are hence solved using an unfit method, leading to suboptimal results. We call them "hidden"

✉ Gaël Mondonneix
gaelmondonneix@yahoo.fr

[1] University of French Polynesia, Géopôle du Pacifique Sud EA4238, LABEX CORAIL, Punaauia, French Polynesia

problems for convenience. For example, an ordinal classification problem (which consists in distributing items into ordered classes) is often solved as a simple multi-class classification problem (which consists in distributing items into classes that do not exhibit an order relation), and it has been shown that it leads to lower performances (see for example [16], showing it in the case of ovarian cancer). This decrease of performance is explained by an improper learning bias: the learning bias implemented by ordinal classification is stronger than the one implemented by multi-class classification (which ignores the ordinal information lying in the data), allowing a better generalization when the classes are ordered.

A learning bias is a set of assumptions which are made explicitly, by setting some parameters, or implicitly by the choice of the method, about the data, and that allow generalizing to unseen instances what has been learned from the training examples ([17] p.42). It can be characterized with respect to two criteria, strength and correctness ([17] p.45): stronger learning biases have a greater power of generalization to new instances; more correct learning biases yield more correct predictions. In that sense, the choice of the learning bias is a crucial aspect of a learning process.

Hidden problems arise mainly in ordinal learning, where ordinal information is not always considered the right way. Ordinal learning stands at the crossroad of classification and regression: addressing discrete (as for classification) but order-related (as for regression) variables. A parallel could be drawn with the levels of measurement [25]: nominal learning, ordinal learning, and metric (interval or ratio) learning.

Depending on the specific problem to solve, ordinal learning can be broken down into ordinal classification and ranking. Ordinal classification is like a traditional multi-class classification task, with the difference that classes are ordered; ranking consists in learning a total order on a set of items. In [8], the authors propose a subdivision of ranking tasks in a) label-ranking, b) instance-ranking and c) object-ranking.

a) Given a set of labels and a set of items, each of which being associated to a permutation on the set of labels, label ranking consists in learning a function that takes as input an item and returns a permutation on the set of labels. For example, a movie advisor system takes a set of users (as items) represented by some features (such as age, gender, etc.) and a set of movies (as labels); learning from the way some users sort the movies, its task is to predict how some other users would sort the same movies, so that it can make recommendations to them regarding their first-ranked movies.
b) Instance-ranking consists in learning a ranking function from a set of items, a set of ordered classes, and a mapping from the items set onto the classes. For example, if movies are represented by some features (such as type, date, nationality, casting, etc.) and a user rates them from one to five stars, then instance-ranking consists in learning from this user-related information a total ordering of the movies that allows to predict the rank of a new movie from its features.
c) Object-ranking consists in learning the same function as for instance-ranking but from a set of partially ordered items. An example of object-ranking can be the same as the one for instance-ranking, with the difference that a movie is not rated with stars but compared with one or more other movies to form a partial order ("movie A is preferred to movie B", "movie C is preferred to movies D and E", etc.). As for ordinal classification problems, which can be "hidden" by multi-class classification problems, ranking problems can be hidden by other problems. The present article focuses on hidden object-ranking problems.

A hidden object-ranking problem (HORP) is an object-ranking problem stated in the form of another learning problem, mostly as a classification problem or as an instance-ranking problem: the training set consists of a set of items together with their corresponding classes, the classes being ordered, and the goal consists in predicting the class of a new item without contradicting the training data (like in classification problems), or in finding a total order on the set of items (like in instance-ranking problems). However, contrarily to a classification or an instance-ranking problem, the class has no intrinsic meaning in a HORP and only represents a way to express a partial binary relation on the set of items: sharing a common class does not imply sharing a common property. In a HORP, the number and configuration of classes have no importance; only the partial order they draw matters.

There exists so far no dedicated method for solving HORPs as object-ranking problems and they are solved as other kinds of problems (mostly as ordinal classification problems or instance-ranking problems), i.e. with an improper learning bias, which is likely to result in lower performances.

The following of this article is structured as follows: in Section 2, we review the different methods for learning ordinal data; in Section 3 we formalize HORPs and propose a kernel machine able to solve them as well as a way to adapt this kernel machine to existing optimization libraries; Section 4 describes an experiment aiming to test the proposed kernel machine on a dataset relative to Tahitian pearls quality assessment; results are given in Section 5 and discussed in Section 6 before concluding.

2 Related work

Depending on the learning bias to implement, different methods exist. Ordinal classification methods build on the assumption that the items can be represented as instances of ordered classes while ranking methods build on the assumption that the items can be totally ordered.

Ordinal classification Ordinal classification methods consist of adaptations or extensions of binary classification methods.

Every binary classifiers can be adapted to ordinal classification using the data replication method [7], [3]. This method consists in learning a discrete cumulative distribution function: for all classes C_k but the last one, a binary classifier is trained to decide whether an item belongs to the union $\bigcup_{i=1}^{k} C_i$. For predicting the class of a new item, an element is considered of class C_k if for the classifiers it belongs to the union $\bigcup_{i=1}^{k} C_i$ but not to the union $\bigcup_{i=1}^{k-1} C_i$.

Binary classifiers based on regression can be extended to ordinal classification. In binary classification, regression consists in fitting sigmoid functions to the data; for example, the Logistic Regression method consists in fitting the logistic function. This principle has been extended to ordinal classification [26]; for example, the Ordered Logit method [13] is an extension of Logistic Regression. Another ordinal classification method is the Ordered Probit method [14], which is similar to Ordered Logit but uses the cumulative distribution function of the standard normal distribution instead of the logistic function.

Instance-ranking Instance-ranking can be carried out with extended SVMs (Support Vector Machines) learning parallel separating hyperplanes between consecutive ordered classes

[9]. In such case, the set of hyperplanes is chosen so as to maximize the thinnest margin. As the hyperplanes are parallel to each other, a vector exists that is normal to all these hyperplanes. Every point representing an item can be projected onto this vector, figuring out a total order on the set of items. Moreover, as this vector results from the set of separating hyperplanes which maximizes the thinnest margin, it puts emphasis on the classes in the sense that it depends on their configuration.

Non-instance ranking Methods that put no particular emphasis on the class-related information consist of two steps [5]:

1. Learning a binary relation that is not necessarily a total order but that complies with the classes.
2. Searching the total order that lies at a minimal distance from the learned binary relation.

Computing such total order is known as the slater's problem [24], which has been proven to be NP-equivalent [10] and for which solutions can in practice only be approximated [5], [23].

Even though these methods allow retaining the minimal constraints from the classes (namely the pairwise relations between items made explicit by the classes) without necessarily putting more emphasis on them, they do not prevent from finding a total order that put emphasis on the classes; in this sense, their learning bias is weaker than if they were dedicated to object-ranking.

3 A method for solving HORPS

A dedicated method for solving HORPs should reduce as much as possible the impact of the classes on the ranking; in other terms, it should favor the total order which would vary the less if the boundaries between classes were sliding along the ranked items.

Taking into account pairwise intra-class information could help in the sense that all possible total orders could then be considered. However, methods based on an explicit listing of the possible total orders are not tractable since the contribution of a class to the number of possible total orders grows exponentially with the number of items in this class. The goal of an object-ranking approach is then to reduce the impact of classes on the selected total order while relying solely on class-related information.

Among all valid total orders, instead of searching for one that puts emphasis on the classes by maximizing the thinnest margin between them as for instance-ranking (see Fig. 1 for an illustration), we choose to search for one that puts emphasis on the order by maximizing the distance to the closest invalid total order (see Fig. 2 for an illustration).

In the following, we introduce a formalization of this method by describing it in a primal form and deriving from it a dual form that allows the use of kernels. Eventually, we propose a way to solve it with existing optimization libraries.

Primal form Let X be a set of m-dimensional real-valued feature vectors representing items. Let Y be a set of classes and $f : Y \to \mathbb{N}$ be an injective function. Let associate to every $x_i \in X$ its corresponding class $y_i \in Y$. Let $\varphi : \mathbb{R}^m \to \mathbb{R}^l$, $(m, l) \in \mathbb{N}^2$, be a possibly nonlinear function and let define the hypothesis space as $H = \mathbb{R}^l$.

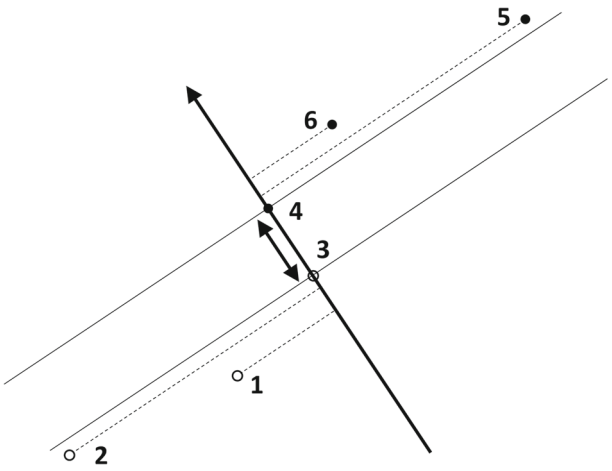


Fig. 1 Illustration of a total order that maximizes the thinnest margin between two classes. The bold double arrow stresses the margin. The bold simple arrow represents the selected vector, on which items are projected in order to form a ranking. The numbers indicate the order found

Let represent a total order by a vector $w \in \mathbb{R}^l$ on which points to be ordered are projected.

Let define the subset of hypotheses consistent with X as $H_X = \left\{ w \in \mathbb{R}^l \;\middle|\; \forall (x_i, x_j) \in X^2, f(y_i) < f(y_j) \Leftrightarrow w^t x_i < w^t x_j \right\}$. If X is a set of observations, then any $w \in H_X$ is an empirically valid hypothesis.

Let choose φ and l such that H_X be non-empty. Then the distance between two total orders can be expressed by the angle between their representative vectors.

A total order that maximizes the distance to the closest invalid total order is represented by the vector $w^* \in \mathbb{R}^l$ that maximizes the smallest angle to a vector representing an invalid

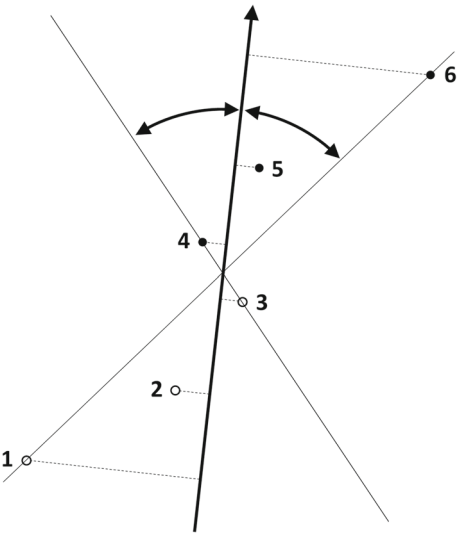


Fig. 2 Illustration of a total order that maximizes the minimal angle to an invalid order. The bold double arrows stress the angle. The bold simple arrow represents the selected vector, on which items are projected in order to form a ranking. The numbers indicate the order found

total order, or similarly, that minimizes the maximal angle to a vector representing a valid total order (Eq. (1)).

$$\begin{aligned} w^* &= argmin_w \left\{ max_{(i,j),f(y_i)<f(y_j)} \angle\big(w, \varphi(x_j) - \varphi(x_i)\big) \right\} \\ &= argmax_w \left\{ min_{(i,j),f(y_i)<f(y_j)} \cos\left(\angle\big(w, \varphi(x_j) - \varphi(x_i)\big)\right) \right\} \\ &= argmax_w \left\{ min_{(i,j),f(y_i)<f(y_j)} \left(\frac{w^t}{\|w\|} \frac{\varphi(x_j)-\varphi(x_i)}{\|\varphi(x_j)-\varphi(x_i)\|} \right) \right\} \end{aligned} \tag{1}$$

Let set $w^t \frac{\varphi(x_j)-\varphi(x_i)}{\|\varphi(x_j)-\varphi(x_i)\|} \geq 1 \quad \forall (i,j), f(y_i) < f(y_j)$ and look for maximizing $\frac{1}{\|w\|}$, which boils down to minimizing $\frac{1}{2}\|w\|^2$ (Eq. (2)).

$$w^* = argmin_w \left\{ \frac{1}{2}\|w\|^2 \quad \middle| \quad w^t \frac{\varphi(x_j) - \varphi(x_i)}{\|\varphi(x_j) - \varphi(x_i)\|} \geq 1 \quad \forall (i,j), f(y_i) < f(y_j) \right\} \tag{2}$$

Dual form We extract now the dual form of the problem by introducing the Lagrangian $L(w,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{(i,j),f(y_i)<f(y_j)} \alpha_{ij} \left(w^t \frac{\varphi(x_j)-\varphi(x_i)}{\|\varphi(x_j)-\varphi(x_i)\|} - 1 \right)$ where $\alpha_{ij} \geq 0$.

The optimization problem consists now of searching for α^*, the optimal vector of coefficients of the training examples (Eq. (3)).

$$\begin{aligned} \alpha^* = \; & argmax_\alpha \Bigg\{ \sum_{(i,j),f(y_i)<f(y_j)} \alpha_{ij} \\ & -\frac{1}{2} \sum_{(i,j),f(y_i)<f(y_j)} \sum_{(k,l),f(y_k)<f(y_l)} \alpha_{ij}\alpha_{kl} \left(\frac{\varphi(x_j)-\varphi(x_i)}{\|\varphi(x_j)-\varphi(x_i)\|} \right)^t \frac{\varphi(x_l)-\varphi(x_k)}{\|\varphi(x_l)-\varphi(x_k)\|} \\ & s.t. \quad \alpha_{mn} \geq 0 \Bigg\} \end{aligned} \tag{3}$$

Introducing slack variables [6] allows enabling this method to tackle noisy datasets (Eq. (4)).

$$\begin{aligned} \alpha^* = \; & argmax_\alpha \Bigg\{ \sum_{(i,j),f(y_i)<f(y_j)} \alpha_{ij} \\ & -\frac{1}{2} \sum_{(i,j),f(y_i)<f(y_j)} \sum_{(k,l),f(y_k)<f(y_l)} \alpha_{ij}\alpha_{kl} \left(\frac{\varphi(x_j)-\varphi(x_i)}{\|\varphi(x_j)-\varphi(x_i)\|} \right)^t \frac{\varphi(x_l)-\varphi(x_k)}{\|\varphi(x_l)-\varphi(x_k)\|} \\ & s.t. \quad 0 \leq \alpha_{mn} \leq C \Bigg\} \end{aligned} \tag{4}$$

In (4), C is the regularization term (preventing overfitting by bounding the complexity of the model). Developing this expression, we obtain Eq. (5).

$$\begin{aligned} \alpha^* = \; & argmax_\alpha \Bigg\{ \sum_{(i,j),f(y_i)<f(y_j)} \alpha_{ij} \\ & -\frac{1}{2} \sum_{(i,j),f(y_i)<f(y_j)} \sum_{(k,l),f(y_k)<f(y_l)} \alpha_{ij}\alpha_{kl} \\ & \frac{\varphi(x_j)^t\varphi(x_l) - \varphi(x_j)^t\varphi(x_k) - \varphi(x_i)^t\varphi(x_l) + \varphi(x_i)^t\varphi(x_k)}{\sqrt{\|\varphi(x_j)\| - 2\varphi(x_j)^t\varphi(x_i) + \|\varphi(x_i)\|}\sqrt{\|\varphi(x_l)\| - 2\varphi(x_l)^t\varphi(x_k) + \|\varphi(x_k)\|}} \\ & s.t. \quad 0 \leq \alpha_{mn} \leq C \Bigg\} \end{aligned} \tag{5}$$

The "kernel trick" allows [22] replacing the product $\varphi(x_i)^t\varphi(x_j)$ by any valid kernel function $K(x_i, x_j)$, that is, by any positive semi-definite kernel function [15].

Let K be such a positive semi-definite kernel function; α^* can be expressed as the (6), where $K(x)$ stands for $K(x, x)$.

$$\begin{aligned}\alpha^* = \; & argmax_\alpha \Bigg\{ \sum_{(i,j),\, f(y_i)<f(y_j)} \alpha_{ij} \\ & -\frac{1}{2} \sum_{(i,j),\, f(y_i)<f(y_j)} \sum_{(k,l),\, f(y_k)<f(y_l)} \alpha_{ij}\alpha_{kl} \\ & \frac{K(x_j,x_l)-K(x_j,x_k)-K(x_i,x_l)+K(x_i,x_k)}{\sqrt{K(x_j)-2K(x_j,x_i)+K(x_i)}\sqrt{K(x_l)-2K(x_l,x_k)+K(x_k)}} \\ & s.t. \quad 0 \leq \alpha_{mn} \leq C \Bigg\}\end{aligned} \tag{6}$$

This is a quadratic programming problem and algorithms exist for solving it (notably the Sequential Minimal Optimization (SMO) method [21], that can solve it in $O(n^3)$, n being the number of inter-class pairs of items).

Adaptation to standard optimization libraries Libraries implementing SMO exist (notably LIBSVM [4]), that can be used for solving the object-ranking optimization problem. But since these libraries are tailored for SVMs, slight modifications are needed to use them for solving our optimization problem. Let derive $X' = \Big\{ x \;\Big|\; \forall (i,j),\, f(y_i) < f(y_j),\, x = (x_{i,1}, x_{i,2}, ..., x_{i,m}, x_{j,1}, x_{j,2}, ..., x_{j,m}) \Big\}$ the set of inter-class pairs of items from the training set.

Let $N' = |X'|$. Let define $l(x)$ and $r(x)$, that return a vector containing respectively the first and the last m values of x. Let define the kernel function K' as in (7), where $K(x)$ stands for $K(x, x)$.

$$K' = \frac{K(l(x_i), l(x_j)) - K(l(x_i), r(x_j)) - K(r(x_i), l(x_j)) + K(r(x_i), r(x_j))}{\sqrt{K(r(x_i)) - 2K(r(x_i), l(x_i)) + K(l(x_i))}\sqrt{K(r(x_j)) - 2K(r(x_j), l(x_j)) + K(l(x_j))}} \tag{7}$$

K' is valid since the operations are done in the feature space on a valid kernel K. The expression for α^* is finally given in (8).

$$\alpha^* = argmax_\alpha \Bigg\{ \sum_{i=1}^{N'} \alpha_i - \frac{1}{2} \sum_{i=1}^{N'} \sum_{j=1}^{N'} \alpha_i \alpha_j K'(i,j) \quad s.t. \quad 0 \leq \alpha_p \leq C \Bigg\} \tag{8}$$

Using K', object-ranking can be solved with SVM solvers by setting all the labels to 1 (or distributing them between 1 and -1 while inverting the ordered pairs accordingly if one-class classification is not allowed). This is tested against classical methods in the following section.

Learnability The proposed method ranks elements by comparing ordered pairs of elements. Ranking by pairwise comparison can be considered as a classification problem operated on the pairs of elements. There is however a difference: even if all the elements are independent, not all the pairs of elements are. An element of the initial learning set, x_i, associated with the label y_i, appears in all couples $\{(x_i, x_j) | y_i < y_j\}$, which become interdependent elements of the new learning set. The traditional theoretical guarantees relating to learnability and generalization error in classification, obtained under the assumption of independence of the elements, could therefore no longer apply.

However, Amini and Usunier [2] showed, based on Janson's inequality [11], that the theoretical guarantees obtained in the context of Rademacher's complexity [12] can be extended to cases of interdependence of data when the dependency structures are known:

the interdependence of the data affects learning by a factor equal to the fractional chromatic number of the dependency graph.

In the context of ranking by comparison on the elements pairwise, where a new learning set is built from the initial set, the dependency graph of the new set is known: each pair of elements from the original set is represented by a vertex, and two vertices are connected by an edge if the couples they represent have a common element. The fractional chromatic number of this graph can therefore be computed; hence bounds on the generalization error exist and can be computed.

4 Experiment

An experiment is conducted on Tahitian black pearls quality assessment, which is yet an unsolved case of hidden object-ranking problem.

In the simplified Polynesian pearl sorting approach, experts sort pearls into four classes, from A to D by decreasing quality. At first glance, it may look as a multi-class classification task; however, observing the quality assessment process reveals that a number of pearls may change class as new pearls are assessed. The sorting process takes place as if the classes were just a compact way to assess pearls quality that complies with the pearls market requirements, but reveals that the experts, as they perform the task, progressively grasp the underlying total order while simultaneously readjusting the previous limit cases.

As such, Tahitian black pearls quality assessment is a typical object-ranking problem stated in terms of a classification problem.

Dataset Tahitian black pearls can be assessed with respect to various criteria (form, color, luster, etc.); we consider here the criterion of luster, which is the way pearls reflect light. The visual aspects of luster have been detailed in [18] and a way to extract the corresponding features out of photographs of pearls proposed in [19]. The latter is used to gather feature vectors from a dataset containing 864 photographs of 54 Tahitian pearls (16 photographs per pearl).

Each instance is represented by 10 features meant to capture the visual aspects of its luster. Roughly, a feature vector aims to capture the following aspects:

- Appearance of specular reflectance (3 features)
- Appearance of contrast between specular reflectance and diffuse reflectance (1 feature)
- Distinctness of the image reflected by the surface of the pearl (1 feature)
- Haze appearing around the zone of specular reflectance (1 feature)
- Appearance of the in-depth reflectance of light through nacre (1 feature)
- Iridescence (1 feature)
- Mean saturation (1 feature)
- Chromaticity variance (1 feature)

The luster quality of each pearl of the dataset is rated by a senior expert with more than 20 years of experience in black pearls trade. Due to the difficulty of obtaining A class luster pearls (which directly go on the market) from professionals, our dataset contains only pearls from classes B to D. Nevertheless, the problem remains a HORP. In order to have balanced classes, pearls are randomly removed from the classes that contain more pearls than the smaller ones until all classes contain the same number of pearls.

Performance evaluation The performance of the proposed approach is evaluated using repeatedly 9-fold cross-validation on 16 disjoint sets of 54 photographs and computing the mean result of correct classification with its standard deviation.

The results are compared with those similarly obtained using the following other methods: multi-class classification (one-vs-one voting scheme with SVM), ordinal classification (replication method with SVM), non-instance ranking (genetic algorithm with population of 100 and mutation rate of 0.2) and instance-ranking (extended SVM). For all methods, results are given with and without feature selection (for the selected features, see [19]). Since the data exhibit quadratic regularities (see [20]), all kernel methods are tested with a quadratic kernel and the genetic algorithm is tested with basis functions ensuring a quadratic expansion.

The significance of a result is evaluated by computing its binomial-test p-value: as a result corresponds to a way of distributing 54 items into 3 balanced classes, its p-value follows a binomial distribution of parameters $n = 54$ and $p = 1/3$ under the null hypothesis. The significance of the difference between two different methods' results is evaluated by computing the t-test p-value of this difference [1]. We set the significance as $p < 0.05$.

For non-instance ranking and object-ranking methods, which embed no principle for determining the class of an instance, the class of a test example is predicted based on its predicted rank as follows: if the predicted rank lies between the ranks of two training examples belonging to a same class, the test example takes this class as label; if the predicted rank is higher than the best training example of a class and lower than the worst of the subsequent class, the test example takes arbitrarily the lower class as label.

5 Results

The main results are presented in Table 1, which displays for each method the percentage of correctly classified instances together with the corresponding standard deviation and p-value for multi-class classification, ordinal classification, non-instance ranking, instance-ranking and object-ranking. The p-values of the differences between results are presented in Table 2 for all features and in Table 3 for selected features. The p-values of the differences between results obtained with and without feature selection are presented in Table 4.

From Table 1, the worst result ($74.7\% \pm 6.5\%$) is found for multi-class classification with all features while the best one ($94.3\% \pm 3.4\%$) is found for object-ranking with selected features. The p-values of all results (all methods, with and without feature selection, Table 1) are less than 0.05.

Table 1 Percentage of correctly classified instances and related standard deviations, together with respective p-values

	All features	Selected features
Multi-Class Classification	**74.7%** $\pm$ 6.5% ($p < 0.01$)	83.6% $\pm$ 6.4% ($p < 0.01$)
Ordinal Classification	80.9% $\pm$ 5.8% ($p < 0.01$)	86.6% $\pm$ 6.1% ($p < 0.01$)
Non-Instance Ranking	89.7% $\pm$ 3.3% ($p < 0.01$)	91.6% $\pm$ 3.8% ($p < 0.01$)
Instance-Ranking	91.3% $\pm$ 3.4% ($p < 0.01$)	92.6% $\pm$ 4.3% ($p < 0.01$)
Object-Ranking	93.6% $\pm$ 3.9% ($p < 0.01$)	**94.3%** $\pm$ 3.4% ($p < 0.01$)

Extrema are set in bold

Table 2 P-values of the differences between results obtained on all features

ALL FEATURES	MCC	OC	NIR	IR	OR
Multi-Class Classification (MCC)	1				
Ordinal Classification (OR)	< 0.01	1			
Non-Instance Ranking (NIR)	< 0.01	< 0.01	1		
Instance-Ranking (IR	< 0.01	< 0.01	0.02	1	
Object-Ranking (OR)	< 0.01	< 0.01	< 0.01	< 0.01	1

Without feature selection (Table 2), the p-values of the differences between these results are all less than 0.05. With feature selection (Table 3), the p-values of the differences between these results are all less than 0.05 but the difference between instance-ranking and non-instance ranking ($p = 0.19$).

For all methods the p-values of the differences between results on all features and results on selected features (Table 4) are all less than 0.05 but in the case of instance-ranking (0.09) and object-ranking (0.33).

6 Discussion

In this study, we focus on hidden object-ranking problems (HORPs). HORPs are object-ranking problems that can misleadingly be solved with improper methods, i.e. methods implementing learning biases that do not fit well the genuine nature of these problems. We propose a method for solving them with an object-ranking's learning bias. The efficiency of the proposed method is tested against that of multi-class classification, ordinal classification, non-instance ranking and instance-ranking methods, with and without features selection (Table 1), and the significance of the results is assessed using p-values (Tables 2, 3 and 4), significance being set as $p < 0.05$.

From Table 1, with or without feature selection, ranking methods yield better results ($91.6\% \pm 3.8\%$ and $89.7\% \pm 3.3\%$ respectively for the worst case) than ordinal classification methods ($86.6\% \pm 6.1\%$ and $80.9\% \pm 5.8\%$ respectively for the best case). Among classification methods, ordinal classification yields better results ($86.6\% \pm 6.1\%$ and $80.9\% \pm 5.8\%$ with and without feature selection respectively) than multi-class classification ($83.6\% \pm 6.4\%$ and $74.7\% \pm 6.5\%$ with and without feature selection respectively). Among ranking methods, object-ranking yields better results ($94.3\% \pm 3.4\%$ and $93.6\% \pm 3.9\%$ with and without feature selection respectively) than instance-ranking ($92.6\% \pm 4.3\%$ and $91.3\% \pm 3.4\%$ with and without feature selection respectively), itself yielding better results

Table 3 P-values of the differences between results obtained on selected features

SELECTED FEATURES	MCC	OC	NIR	IR	OR
Multi-Class Classification (MCC)	1				
Ordinal Classification (OR)	0.02	1			
Non-Instance Ranking (NIR)	< 0.01	< 0.01	1		
Instance-Ranking (IR	< 0.01	< 0.01	0.19	1	
Object-Ranking (OR)	< 0.01	< 0.01	< 0.01	0.02	1

Table 4 P-values of the differences, per method, between results obtained on all features and on selected features

Multi-Class Classification (MCC)	< 0.01
Ordinal Classification (OR)	< 0.01
Non-Instance Ranking (NIR)	< 0.01
Instance-Ranking (IR	0.09
Object-Ranking (OR)	0.33

than non-instance ranking ($91.6\% \pm 3.8\%$ and $89.7\% \pm 3.3\%$ with and without feature selection respectively). From Tables 1, 2 and 3, p-values of the differences between results obtained on all features are all significant but the one between instance and non-instance ranking in the case of feature selection (p-value $= 0.19$).

These results can be interpreted in the lights of the learning biases the different methods implement: as mentioned in Section 1, stronger learning biases have a greater power of generalization to new instances while more correct learning biases yield more correct predictions ([17] p.45).

The performances increase from multi-class classification to ordinal classification and from ordinal classification to ranking. This seems to reflect the strength of the learning biases these methods implement:

- Multi-class classification has the weakest learning bias, that makes no assumption about the relations between items nor between classes.
- Ordinal classification has a stronger learning bias in that sense that even though it makes no assumption about the relations between items, it constraints the classes to be totally ordered.
- Ranking has the strongest learning bias, that constraints the items to be totally ordered (the classes being thus necessarily totally ordered too).

Among the three ranking methods, the performances increase from non-instance ranking to instance-ranking and from instance-ranking to object-ranking. The difference between non-instance ranking and the two other ranking methods can be explained again by the strength of the learning bias. Indeed, a non-instance ranking's learning bias constraints the items to be totally ordered but impose no constraint on the total order itself while the two others do: an instance-ranking's learning bias imposes to choose the total order that emphasizes the most the classes whereas an object-ranking's learning bias imposes to choose the total order that emphasizes them the least. Since instance-ranking and object-ranking have equally strong learning biases (both reduce to only one the number of hypotheses among which the optimal can lie), the strength of the learning bias cannot explain the difference between their performances. Nonetheless, the correctness of their learning biases might: the constraints imposed by the object-ranking's learning bias seem to be more relevant for the problem we tackle than these imposed by the instance ranking's bias. Indeed, it has been reported in Section 4 that the classes have no intrinsic meaning in Tahitian black pearls quality assessment, so it seems reasonable that a method favoring the least class-dependent total order performs better.

The results seem to confirm that the proposed kernel machine implements a learning bias that fits well the hidden object ranking problem of Tahitian pearls quality assessment. They do not allow however to state that this method is optimal for HORPs in general, and further

experiments need to be conducted to investigate how it behaves on other datasets. For example, the dataset we use is relatively small, so an interesting point to investigate is whether the improvement brought by our approach is maintained on bigger datasets. Moreover, the dataset has been shown to exhibit quadratic regularities and as such the quadratic kernel has been used for capturing them; so another interesting point to investigate is whether our method still performs better with other types of regularities.

From Table 4, feature selection improves the results for every method, however the improvement is not significant for instance-ranking (p-value $= 0.09$) and object-ranking (p-value $= 0.33$). The fact that object-ranking results with and without feature selection are very close (respectively $93.6\% \pm 3.9\%$ and $94.3\% \pm 3.4\%$) may be interpreted as a hint that the learning bias embedded in object-ranking fits the problem well. This could allow skipping a preprocessing such as feature selection, thus alleviating the data preprocessing work load.

7 Conclusion

In this article, we focus on Hidden Object-Ranking Problems (HORPs), which are object-ranking problems stated in the form of classification or instance-ranking problems. There exists so far no dedicated algorithm for solving them with the right learning bias and they are usually solved as classification or instance-ranking problems. We propose a method for solving them with an appropriate learning bias, that is, in a way that lessens the impact of the classes on the results. This method is formalized, a kernel machine derived and formulation is proposed that allows implementing this kernel machine with traditional optimization libraries. We apply our method to a real world HORP, Tahitian black pearls quality assessment, and show that it yields better results ($93.6\% \pm 3.9\%$ of correct predictions without feature selection, $94.3\% \pm 3.4\%$ with feature selection) than the best of the other tested methods ($91.3\% \pm 3.4\%$ and $92.6\% \pm 4.3\%$ without and with feature selection for the instance-ranking approach), this improvement being significant (p-value < 0.05).

In a future work, it could be interesting to further investigate the difference between instance-ranking and object-ranking in a broader scope than HORPs.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Altman DG (1991) Statistics in medicine. https://doi.org/10.1002/sim.4780101015
2. Amini MR, Usunier N (2015) Learning with partially labeled and interdependent data. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-15726-9
3. Cardoso JS, Sousa R (2012) Measuring the Performance of Ordinal Classification. Int J Pattern Recognit Artif Intell 25(08):1173. https://doi.org/10.1142/s0218001411009093
4. Chang CC, Lin CJ, Tieleman T (2008) LIBSVM : A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST) 307:1. https://doi.org/10.1145/1961189.1961199

5. Cohen WW, Schapire RE, Singer Y (1999) Journal of Artificial Intelligence Research. https://doi.org/10.1613/jair.587
6. Cortes C, Vapnik V (1995) In Silico Log P Prediction for a Large Data Set with Support Vector Machines, Radial Basis Neural Networks and Multiple Linear Regression. Chem Biol Drug Des 20:273. https://doi.org/10.1111/j.1747-0285.2009.00840.x. http://www.ncbi.nlm.nih.gov/pubmed/19549084
7. Frank E, Hall M (2001) A Simple Approach to Ordinal Classification. In: Proceedings of the 12th european conference on machine learning. Springer, Berlin, pp 145–156. http://dl.acm.org/citation.cfm?id=645328.649997
8. Fürnkranz J, Hüllermeier E (2010) Preference Learning: An Introduction. In: Preference learning. Springer, Berlin, pp 1–17, https://doi.org/10.1007/978-3-642-14125-6. http://www.ke.tu-darmstadt.de/publications/papers/PLBook-Introduction.pdf
9. Herbrich R, Graepel T, Obermayer K (1999) Science. https://doi.org/10.1049/cp:19991091
10. Hudry O (2010) On the complexity of Slater's problems. Eur J Oper Res 203(1):216. https://doi.org/10.1016/j.ejor.2009.07.034
11. Janson S (2004) Random structures and algorithms. https://doi.org/10.1002/rsa.20008
12. Koltchinskii V, Panchenko D (2000) In high dimensional probability II. https://doi.org/10.1007/978-1-4612-1358-1-29
13. McCullagh P (1980) Regression Models for Ordinal Data. Journal of the Royal Statistical Society: Series B (Methodological) 42(2):109. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x
14. McKelvey RD, Zavoina W (1975) A statistical model for the analysis of ordinal level dependent variables. J Math Sociol 4(1):103. https://doi.org/10.1080/0022250X.1975.9989847
15. Mercer J (1909) Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. Philos T R Soc A 209(441-458):415. https://doi.org/10.1098/rsta.1909.0016. http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.1909.0016
16. Misganaw B, Vidyasagar M (2015) Exploiting Ordinal Class Structure in Multiclass Classification: Application to Ovarian Cancer. IEEE Life Sciences Letters 1(1):15. https://doi.org/10.1109/lls.2015.2451291
17. Mitchell T (1997) Machine learning. McGraw-Hill Science. https://doi.org/10.1007/BF00116892
18. Mondonneix G, Chabrier S, Mari JM, Gabillon A, Barriot JP (2017) Irish pattern recognition & classification society. In: McDonald J, Markham C, Winstanley A (eds) Proceedings of the 19th irish machine vision and image processing conference, Maynooth, pp 186–193
19. Mondonneix G, Chabrier S, Mari JM, Gabillon A (2017) In proceedings of the IEEE applied imagery pattern recognition workshop: big data, analytics and beyond
20. Mondonneix G, Chabrier S, Mari JM, Gabillon A (2018) Ordinal Learning with Vector Space Based Binary Predicates and Its Application to Tahitian Pearls' Luster Automatic Assessment. In: Lecture notes in computer science. Springer, Cham, pp 90–98
21. Platt J (1998) http://Citeseer.ist.psu.edu/platt98sequential.html, pp 1–21
22. Segaran T (2007) Programming collective intelligence, pp 197–224
23. Shmoys D (1996) Cut problems and their application to divide and conquer. In: Approximation algorithms for NP-hard problems. PWS Publishing Co., pp 192–235
24. Slater P (1961) Biometrika 48(3/4):303. https://doi.org/10.2307/2332752
25. Stevens SS (1946) Sci 103(2684):677 LP. http://science.sciencemag.org/content/103/2684/677.abstract
26. Winship C (1984) Regression models with ordinal variables. Am Sociol Rev 49:512. http://www.jstor.org/stable/2095465%5Cnpapers2://publication/uuid/9DE79234-EA1B-4D8A-9B7F-2E6006D34714

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dr. Gaël Mondonneix has been working on Machine Learning applied to Tahitian Black Pearls quality assessment. His research focuses on Classification and Ranking problems.



Dr. Jean Martial Mari is Associate Professor in Computer Science at the University of French Polynesia working on signal and image processing. In year 2000 he received the Electrical Engineering diploma of the INSA Lyon, France, with a specialization in Signal Processing, and a Master degree (DEA) in signal and image processing. In 2004 he received his PhD in Acoustics and in 2012 his French "Habilitation to Lead Research" (HDR) from the Université Claude Bernard Lyon 1 (France).



Dr. Sébastien Chabrier is currently associate professor in Computer Science at the University of French Polynesia, working and specialized on image processing and machine learning.



Pr. Alban Gabillon has been doing most of his research works on various aspects of Computer Security. Recently he began to explore the fields of Image Processing and Machine Learning.