Playing the Large Margin Preference Game

Playing the Large Margin Preference Game

Mirko Polato[1], Guglielmo Faggioli[1], Ivano Lauriola[1,2], and Fabio Aiolli[1]

[1] University of Padova, Department of Mathematics, Italy
{mpolato,gfaggiol,aiolli}@math.unipd.it
[2] Fondazione Bruno Kessler, Trento, Italy
lauriola@phd.unipd.it

Abstract. We propose a large margin preference learning model based on game theory to solve the label ranking problem. Specifically, we show the proposed formulation is able to perform multiclass classification by solving a single convex optimization problem. Generally, such formulation, although theoretically well-founded, requires to learn a large number of parameters. To reduce the computational complexity, we propose a strategy based on the solution of smaller subproblems, that can be further optimized by exploiting techiniques borrowed from multi-armed bandits literature. Finally, we show how the proposed framework exhibits state-of-the-art results on many benchmark datasets.

Keywords: game theory · svm · large margin · kernel method · large scale

1 Introduction

For many years, Support Vector Machine (SVM) has been one of the most studied and heavily used Machine Learning (ML) method. Besides its state-of-the-art performance in many learning tasks, its success is mainly due to its theoretical foundation. SVM roots in statistical learning theory [20] and follows the principle of structural risk minimization to control the generalization ability of a learning machine. It belongs to the family of large margin models and its elegant formulation makes it suitable for connections with other theoretical fields. An example being its strong relation with game theory (GT). For instance, it is well known that hard margin SVM can be cast into a two-players zero-sum game [1]. GT has also been related to other ML techniques, including, boosting [8] and linear regression [13]. More recently, similar connections have been made between Preference Learning (PL) and GT [16].

Starting from this last finding, we present a theoretically well-founded preference learning framework inspired by game theory for multi-class classification problems. Specifically, we define a (generalized) linear PL model in which the large margin problem is cast into a two-players zero-sum game. The proposed framework is general enough to be easily used with kernel in order to handle non-linear problems. We show how this model can be trained by solving a simple convex optimization problem. However, akin other kernel methods, like SVM, it could not be suited for large scale problems. To this regard we also propose a technique inspired by multi-armed bandits to speed up the learning process.

The remainder of the paper is structured as follows: Section 2 introduces all the necessary background. Section 3 and 4 describe the main contributions of the paper. Finally, Section 5 and 6 show the experimental assessment and discuss possible future research directions.

2 Background

2.1 Preference Learning

Preference learning (PL) is a sub-task in machine learning in which the input data consists of preference relations. In PL problems, the goal is to construct a preference model able to predict preferences for previously unseen items. The typical assumption is that preferences are in agreement with some utility function g_θ. The task then becomes to find the parameters θ of the utility function g.

Label ranking is one of the main PL tasks [10]: given a set of input patterns $\mathbf{x}_i \in \mathcal{X}$, $i \in [1, \dots, n]$, and a finite set of labels $\mathcal{Y} \equiv \{y_1, y_2, \dots, y_m\}$ the goal is to learn the utility function $g_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which assigns a fitness score for each instance-label pair $(\mathbf{x}, y)$. Label ranking represents a generalization of a classification task, since, given an instance $\mathbf{x}$, g_θ implicitly defines a total order over $\mathcal{Y}$. In the label ranking context, the training set consists of pairwise preferences $y_i \succ_{\mathbf{x}} y_j$, $i \neq j$, i.e., for the pattern $\mathbf{x}$, y_i is preferred to y_j. In the special case of classification, in which $\mathbf{x}$ is associated to a unique label y_i, the preferences' set is

$$\{y_i \succ_{\mathbf{x}} y_j \mid 1 \leq j \neq i \leq m\}.$$

In this work we focus on (generalized) linear preference models [19, 2] on some feature space $\mathcal{F}$ induced by an embedding function ϕ, i.e., $g_{\mathbf{w}}(\phi(\mathbf{x}), y) = \mathbf{w}^\intercal \psi(\phi(\mathbf{x}), y)$, where $\mathbf{w}$ is the parameters vector, ψ is a joint representation of instance-label pairs, and $\phi : \mathcal{X} \to \mathbb{R}^d$ is the embedding function.

Since the preferences are ranked according to the utility function, given a preference $y_i \succ_{\mathbf{x}} y_j$ then $g_{\mathbf{w}}(\phi(\mathbf{x}), y_i) > g_{\mathbf{w}}(\phi(\mathbf{x}), y_j)$ should hold, and thus

$$\mathbf{w}^\intercal \psi(\phi(\mathbf{x}), y_i) > \mathbf{w}^\intercal \psi(\phi(\mathbf{x}), y_j) \Rightarrow \mathbf{w}^\intercal (\psi(\phi(\mathbf{x}), y_i) - \psi(\phi(\mathbf{x}), y_j)) > 0,$$

which can be interpreted as the *margin* (or *confidence*) of the preference.

The instance-label joint representation used in this work is based on the Kesler's construction for multi-class classification [15, 6, 11]. That is a very powerful tool for extending learning algorithms for binary classifiers to the multi-class setting. The Kesler's construction allows, by using an appropriate instances' representation, to solve multi-class problems using a single linear function instead of decomposing them into many binary sub-problems. The construction can be formalized as in the following.

Given an instance (possibly embedded in a feature space) $\phi(\mathbf{x})$ with label y, we define the instance-label representation $\psi : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}^{d \cdot m}$ as $\psi(\phi(\mathbf{x}), y) =$

$\mathbf{e}^m_y \otimes \phi(\mathbf{x})$, where the symbol $\otimes$ indicates the Kronecker product and $\mathbf{e}^m_y$ is the y-*th* canonical basis of $\mathbb{R}^m$:

$$\psi(\phi(\mathbf{x}),y) = \mathbf{e}^m_y \otimes \phi(\mathbf{x}) = (\,\underset{\underset{1}{\uparrow}}{\mathbf{0}}\,;\,\underset{\underset{2}{\uparrow}}{\mathbf{0}}\,;\ldots;\,\underset{\underset{y}{\uparrow}}{\phi(\mathbf{x})}\,;\mathbf{0};\ldots;\,\underset{\underset{m}{\uparrow}}{\mathbf{0}}\,) \in \mathbb{R}^{d\cdot m},$$

where $\mathbf{0}$ are d-dimensional zero vectors. Therefore, given a preference $y_i \succ_{\mathbf{x}} y_j$ we construct its corresponding embeddings $\mathbf{z} \in \mathbb{R}^{d\cdot m}$ as

$$\begin{aligned} \mathbf{z} &= \psi(\phi(\mathbf{x}),y_i) - \psi(\phi(\mathbf{x}),y_j) = (\mathbf{e}^m_{y_i} - \mathbf{e}^m_{y_j}) \otimes \phi(\mathbf{x}) \\ &= (\mathbf{0};\ldots;\,\underset{\underset{y_i}{\uparrow}}{\phi(\mathbf{x})}\,;\mathbf{0};\ldots;\,\underset{\underset{y_j}{\uparrow}}{-\phi(\mathbf{x})}\,;\mathbf{0};\ldots;\mathbf{0}) \in \mathbb{R}^{d\cdot m}. \end{aligned}$$

At prediction time, given a new instance $\mathbf{x}_{\text{new}}$, labels are ranked according to the score $g_{\mathbf{w}}(\phi(\mathbf{x}_{\text{new}}),y)$, $\forall y \in \mathcal{Y}$. In case of classification, the predicted label for $\mathbf{x}_{\text{new}}$ is $\hat{y} = \arg\max_{y\in\mathcal{Y}} g_{\mathbf{w}}(\phi(\mathbf{x}_{\text{new}}),y)$.

2.2 Game Theory

Game theory is the science of strategic reasoning that studies the behaviour of rational game players who are trying to maximize their utility. Specifically, in this paper, we focus on finite two-players zero-sum games. The strategic form of a two-players zero-sum game is defined by a triplet (P, Q, M), where P and Q are finite non-empty set of strategies for player $\mathtt{P}$ and $\mathtt{Q}$, respectively, and $M : P \times Q \to \mathbb{R}$ is a function that associates a value $M(i,j)$ to each pair of strategies (i,j) s.t. $i \in P, j \in Q$. $M(i,j)$ represents the payoff of $\mathtt{Q}$ and the loss of $\mathtt{P}$. Since P and Q are finite sets, M can be represented as a matrix $\mathbf{M} \in \mathbb{R}^{|P|\times|Q|}$, called payoff matrix (or game matrix), such that $\mathbf{M}_{ij} = M(i,j)$, where $|P|$ and $|Q|$ are the number of available strategies for $\mathtt{P}$ and $\mathtt{Q}$, respectively. Each matrix entry $\mathbf{M}_{i,j}$ represents the loss of $\mathtt{P}$, or equivalently the payoff of $\mathtt{Q}$, when the strategies i and j are played by the players. The game takes place in rounds in which the two players play simultaneously: the row player ($\mathtt{P}$) picks a row $p \in P$, and the column player ($\mathtt{Q}$) picks a column $q \in Q$ of $\mathbf{M}$. The goal of the player $\mathtt{P}$ is to define a strategy that minimizes its expected loss V. Conversely, the player $\mathtt{Q}$ aims at finding a strategy that maximizes its payoff. Players strategies are typically represented as stochastic vectors $\mathbf{p} \in \mathscr{S}_P$ and $\mathbf{q} \in \mathscr{S}_Q$, respectively, where $\mathscr{S}_P = \{\mathbf{p} \in \mathbb{R}^{|P|}_+ \mid \|\mathbf{p}\|_1 = 1\}$ and $\mathscr{S}_Q = \{\mathbf{q} \in \mathbb{R}^{|Q|}_+ \mid \|\mathbf{q}\|_1 = 1\}$. It is well known [14] that the best pair of optimal strategies $(\mathbf{p}^*, \mathbf{q}^*)$, i.e., the saddle-point (or Nash equilibrium) of $\mathbf{M}$, can be computed by

$$V^* = \mathbf{p}^{*\intercal}\mathbf{M}\mathbf{q}^* = \min_{\mathbf{p}} \max_{\mathbf{q}} \mathbf{p}^\intercal\mathbf{M}\mathbf{q} = \max_{\mathbf{q}} \min_{\mathbf{p}} \mathbf{p}^\intercal\mathbf{M}\mathbf{q},$$

where V^* is the value of the game.

3 PL maximal margin as a two-players zero-sum game

In Section 2.1 we have introduced the concept of margin of a preference. Akin classical classification scenarios [17], also in PL contexts large margins correspond to good generalization capability of the ranker [1].

As mentioned previously, we consider a hypothesis space $\mathcal{H}$ composed by linear functions, i.e., $\mathcal{H} \equiv \{\mathbf{z} \mapsto \mathbf{w}^\intercal \mathbf{z} \mid \mathbf{w}, \mathbf{z} \in \mathbb{R}^{d \cdot m}\}, \|\mathbf{w}\|_2 = 1\}$. Given a hypothesis $\mathbf{w}$, we say that $\mathbf{w}$ satisfies a preference $\mathbf{z}$ if its margin is strictly positive, that is, iff $\rho_{\mathbf{w}}(\mathbf{z}) = \mathbf{w}^\intercal \mathbf{z} > 0$. We assume to have a set of training preferences of the form $\mathcal{T} \equiv \{(y_+ \succ_{\mathbf{x}} y_-)\}, |\mathcal{T}| = n(m-1)$ which can be easily transformed to their corresponding vectorial representation as previously described. According to the maximum margin principle, we aim to select $\mathbf{w}$ such that it maximizes the minimum margin over the training preferences. Following the line of [1, 16], we cast the margin maximization problem into a two-players zero-sum game. Specifically, let $Q \equiv \mathcal{H}$, and let $P \equiv \mathcal{T}$ be the set of strategies for the player `Q` and `P`, respectively. The game takes place in rounds, where `Q` selects an hypothesis $\mathbf{w} \in \mathcal{H}$ and `P` selects a preference $\mathbf{z}$ from $\mathcal{T}$. `Q` wants to maximize its payoff, which is the margin achieved by $\mathbf{w}$ on $\mathbf{z}$. Conversely, `P` aims to minimize its loss by defining a mixed strategy over the set of training preferences, which can be seen as a probability distribution $\mathbf{p} \in \mathscr{S}_P$ over the preferences. The value of this game, i.e., the expected margin, is computed by solving

$$V^* = \min_{\mathbf{p}} \max_{\|\mathbf{w}\|_2=1} \mathbb{E}_{\mathbf{p}} \left[\rho_{\mathbf{w}}(\mathbf{z})\right] = \min_{\mathbf{p}} \max_{\|\mathbf{w}\|_2=1} \sum_{i=1}^{|P|} p_i \rho_{\mathbf{w}}(\mathbf{z}_i) \tag{1}$$

$$= \min_{\mathbf{p}} \max_{\|\mathbf{w}\|_2=1} \sum_{i=1}^{|P|} p_i \mathbf{w}^\intercal \mathbf{z}_i = \min_{\mathbf{p}} \max_{\|\mathbf{w}\|_2=1} \mathbf{w}^\intercal \left(\sum_{i=1}^{|P|} p_i \mathbf{z}_i \right). \tag{2}$$

It is well known that the unitary norm maximizer of Eq. (2) is

$$\mathbf{w} \propto \sum_{i=1}^{|P|} p_i \mathbf{z}_i = \mathbf{Z}^\intercal \mathbf{p},$$

where $\mathbf{Z} \in \mathbb{R}^{|P| \times (d \cdot m)}$ is the matrix with the preference embeddings arranged in the rows, and hence we can rewrite Eq. (2) as

$$V^* = \min_{\mathbf{p}} \sum_{i=1}^{|P|} p_i \sum_{j=1}^{|P|} p_j \mathbf{z}_i^\intercal \mathbf{z}_j = \min_{\mathbf{p}} \mathbf{p}^\intercal \mathbf{K}_{\mathbf{z}} \mathbf{p}, \tag{3}$$

where $\mathbf{K}_{\mathbf{z}} \in \mathbb{R}^{|P| \times |P|}$ is a kernel matrix between preferences, that is $\mathbf{K}_{\mathbf{z}}[i,j] = \mathbf{z}_i^\intercal \mathbf{z}_j$. Given the Kesler's construction described in Section 2.1, then $\mathbf{K}_{\mathbf{z}}$ can be computed as:

$$\mathbf{K}_{\mathbf{z}}[i,j] = (\mathbf{0}, \ldots, \underset{\substack{\uparrow \\ y_i^+}}{\phi(\mathbf{x}_i)}; \ldots; \underset{\substack{\uparrow \\ y_i^-}}{-\phi(\mathbf{x}_i)}; \ldots; \mathbf{0})^\intercal (\mathbf{0}; \ldots; \underset{\substack{\uparrow \\ y_j^+}}{\phi(\mathbf{x}_j)}; \ldots; \underset{\substack{\uparrow \\ y_j^-}}{-\phi(\mathbf{x}_j)}; \ldots; \mathbf{0})$$

$$= (\llbracket y_i^+ = y_j^+ \rrbracket + \llbracket y_i^- = y_j^- \rrbracket - \llbracket y_i^+ = y_j^- \rrbracket - \llbracket y_i^- = y_j^+ \rrbracket)\kappa(\mathbf{x}_i, \mathbf{x}_j),$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\intercal \phi(\mathbf{x}_j)$ is the kernel function induced by ϕ, and $\llbracket \cdot \rrbracket$ is the indicator function. Equation (3) shows that it is possible to learn the maximal margin hypothesis in the preference space by solving a quadratic optimization problem. This formulation allows to solve a multi-class classification problem without the need of decomposing it in multiple binary classification problems. However, when the number of preferences is huge computing (3) on the whole kernel matrix $\mathbf{K_z}$ can be prohibitive. For this reason in the next section we provide a technique for efficiently approximating the value of the game, and thus learning the model.

4 Approximating the value of the PL game

There is a large body of research in the game theory community which deals with the problem of approximating the value of the game for huge game matrices [9, 7, 4, 3, 5]. However, such techniques assume the availability of the whole game matrix which is not always feasible in our context. More recently [16], an incremental approach for solving large game matrices w.r.t. the number of columns has been proposed in which only a budget of columns are considered at each iteration. Unfortunately, limiting the number of columns only could not be enough when the number of preferences is huge. For this reason, we propose a method that approximates the value of the game (as well as the strategies of the players) by combining the solutions of many sub-games that consider only squared sub-matrices of the whole game matrix $\mathbf{K_z}$.

Specifically, let T be the number of sub-games we want to solve, and let $\boldsymbol{\Pi}_t \in \{0, 1\}^{|P| \times s}$ be the selection matrix used to select rows/columns from $\mathbf{K}_z$ for the t-th game. Thus, each sub-game matrix $\mathbf{K}_t \in \mathbb{R}^{s \times s}$ $(s \ll |P|)$ can be obtained as $\mathbf{K}_t = \boldsymbol{\Pi}_t^\intercal \mathbf{K_z} \boldsymbol{\Pi}_t$. Let $\hat{\mathbf{p}}_t$ be the optimal strategy for the t-th sub-game, then we can project back the solution by computing $\mathbf{p}_t = \boldsymbol{\Pi}_t \hat{\mathbf{p}}_t$

Once all $\mathbf{p}_t$ have been computed, we aim to combine these sub-strategies in order to get a strategy for the whole game. The best convex combination of the $\mathbf{p}_t$'s can be achieved by solving the following convex optimization problem

$$\boldsymbol{\alpha}^* = \min_{\boldsymbol{\alpha} \in \mathscr{S}^T} \boldsymbol{\alpha}^\intercal \left(\mathbf{P}^\intercal \mathbf{K_z} \mathbf{P}\right) \boldsymbol{\alpha} = \min_{\boldsymbol{\alpha} \in \mathscr{S}^T} \boldsymbol{\alpha}^\intercal \mathbf{G} \boldsymbol{\alpha}, \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{T \times |P|}$ is the matrix where the strategies ($\mathbf{p}_t$) of the sub-games are arranged in the rows, and $\mathbf{G} = \mathbf{P}^\intercal \mathbf{K_z} \mathbf{P} \in \mathbb{R}^{T \times T}$. Clearly, the value of the game $\widetilde{V} = \boldsymbol{\alpha}^* \mathbf{G} \boldsymbol{\alpha}^*$ is an approximation of V^* and specifically $V^* \leq \widetilde{V}$

From the formulation given in (4) it seems that it is still necessary to compute the whole kernel matrix $\mathbf{K_z}$. However, it can be observed that since $\mathbf{P}$ is built upon the best strategies of the sub-games, in each row at most s entries are non zero. Hence, computing $\mathbf{G}$ can be highly optimized, e.g., by computing each row individually. Nevertheless, when the number of preferences is particularly large computing $\mathbf{G}$ remains computationally expensive. Anyhow, it is possible to get a reasonable approximation avoiding to solve the optimization problem by fixing $\boldsymbol{\alpha}^*$ to the uniform distribution, which corresponds to computing the average over $\mathbf{p}_t$.

4.1 Sub-game selection strategy

Even though $\widetilde{V}$ is the best we can achieve from the combination of the partial strategies $\mathbf{p}_t$, the sub-game selection plays a key role to get good value of the game with the proposed method.

A naïve way of computing the sub-game matrix is by randomly drawing rows/columns from the uniform distribution. This strategy has the advantage of being highly parallelizable, since each sub-game can be solved independently.

Borrowing from the reinforcement learning literature, we propose a generalization of the strategy presented above in which samples are randomly drawn from a distribution that depends on the solution of the previous sub-games. The main idea is to iteratively adjust the distribution according to how much the previously selected preferences (i.e., strategies) contributed to the mixed-strategy (i.e., their weight in the hypothesis). Specifically, let $\mathbf{d}_t \in \mathscr{S}^{|P|}$ be probability distributions over all the training preferences at iteration t, and let $\mathbf{d}_1 = \mathbf{1}\frac{1}{|P|}$ be the uniform distribution over all preferences. At iteration $t+1$ a new random sample of preferences is drawn accordingly to $\mathbf{d}_{t+1}$ which is defined as

$$\mathbf{d}_{t+1} = (1-\lambda)\mathbf{d}_t + \lambda \mathbf{p}_t$$

where $0 \leq \lambda \leq 1$, and $\mathbf{p}_t$ is the solution of the t-th game as in Section 4. Essentially, λ defines how much the previous strategies influence the sampling distribution for the next games. $\lambda = 0$ means that the previous games have no influence in the next sampling. Conversely, $\lambda = 1$ indicates that all random samples will be drawn according to $\mathbf{p}_1$ (i.e., the solution of the first sub-game). In other words λ is a trade-off between exploration ($\lambda \to 0$) and exploitation ($\lambda \to 1$), with a similar effect of ϵ in the ϵ-greedy algorithm [18] for the multi-armed bandit problem.

Finally, the ranker hypothesis is computed as a combination over all $\mathbf{p}_t$, that is

$$\mathbf{w} \propto \left[\sum_{t=1}^{T} \alpha_t \mathbf{p}_t\right]^{\mathsf{T}} \mathbf{Z},$$

where $\boldsymbol{\alpha}$ can be optimized as in Eq. (4), or fixed, for example, to the unifrom distribution. Algorithm 1, dubbed LMPG (Large Margin Preference Game), provides the pseudo-code of the method just described. In the following we will indicate with LMPG* the algorithm when $s = |P|$, LMPG-α when $\boldsymbol{\alpha}$ is optimized according to (4), and with LMPG when $\boldsymbol{\alpha}$ is fixed to the uniform distribution.

5 Experiments

In this section, experiments done to empirically evaluate techniques described in Section 3 and 4 are presented.

Algorithm 1: LMPG: Large Margin Preference Game

```
Input:
    $P$: set of training preferences
    $s$ : sample size
    $\lambda$: exploration-exploitation trade-off hyper-parameter
    $T$: number of iterations
Output:
    $\widetilde{\mathbf{w}}$: preference ranking model
$\mathbf{d}_1 \leftarrow \mathbf{1}\frac{1}{|P|}$
for $t \leftarrow 1$ to $T$ do
    $Q \leftarrow$ random sampling (w/o replacement) over $P$ of $s$ preferences
      according to $\mathbf{d}_t$
    $\mathbf{K}_t \leftarrow$ kernel matrix s.t. $\mathbf{K}_t[i,j] = \mathbf{z}_i^\intercal \mathbf{z}_j, \forall \mathbf{z}_i, \mathbf{z}_j \in Q$
    $\mathbf{p}_t \leftarrow \min_{\mathbf{p}} \mathbf{p}^\intercal \mathbf{K}_t \mathbf{p}$
    $\mathbf{d}_{t+1} \leftarrow (1-\lambda)\mathbf{d}_t + \lambda \mathbf{p}_t$
end
computing $\boldsymbol{\alpha}$ (e.g., by means of (4))
$\overline{\mathbf{p}} \leftarrow \sum_{t=1}^{T} \alpha_t \mathbf{p}_t$
$\widetilde{\mathbf{w}} \leftarrow \sum_{i=1}^{|P|} \overline{p}_i \mathbf{z}_i, \mathbf{z}_i \in P$
return $\widetilde{\mathbf{w}}$
```

The proposed techniques have been evaluated on five different publicly available datasets:

`tic-tac-toe` is a dataset containing 958 ending positions of the game tic-tac-toe, and the task is to classify whether the × is the winner;

`breast-cancer` is the well known Breast Cancer Wisconsin Diagnostic Dataset, where the task is to classify a tumor as malignant or benign. For more details about the dataset please refer to [12];

`mnist-49` `mnist` is a (well known) dataset of handwritten digits. We extracted from it a single classification task which consists in classifying the digit 4 against the digit 9;

`segment` This dataset is an image segmentation database. 7 outdoor images are possible istances and images have been randomly selected. The images were handsegmented to create a classification for every pixel. Each instance is a 3x3 region.

`w8a` Dataset used for fast training of support vector machines using sequential minimal optimization.

Table 1 summarizes the information of the selected datasets. Note that, since `segment` is a multiclass dataset, the number of preferences correspond to the number of examples, multiplied the number of classes (minus 1), for a total of 11088 preferences.

All experiments concerning the LMPG method have been carried out using the same procedure. We set $T = 500$, λ have been tested in the set of values

dataset	# classes	training set size	test set size	# features
tic-tac-toe	2	766	192	27
breast-cancer	2	545	137	90
mnist-49	2	11025	2757	779
segment	7	1848 (11088)	462	19
w8a	2	39799	9950	300

Table 1. Datasets information: number of classes, training set and test set size, and number of features. In parenthesis the corresponding number of preferences. When not indicated the number of preferences is equal to the number of examples.

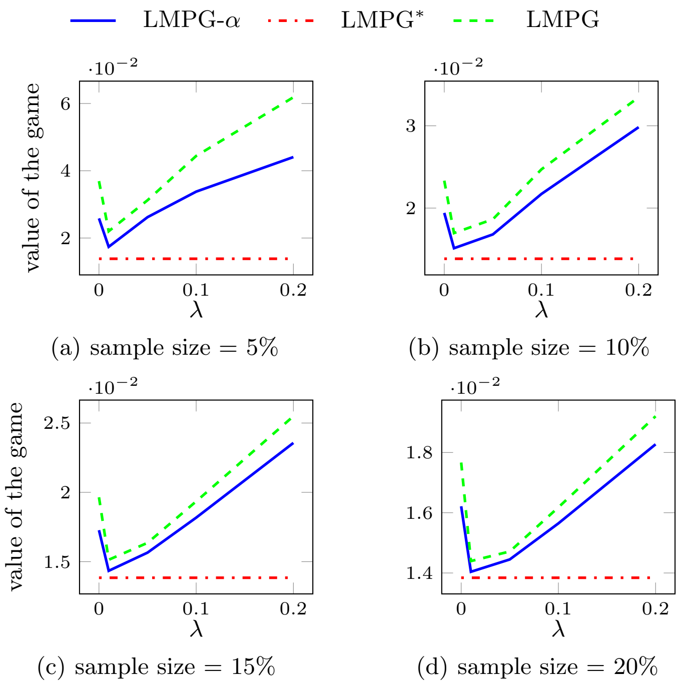
	method	hyper-parameters	accuracy	precision	recall	F1
breast	LMPG*		0.9635	0.9625	0.9585	0.9605
	LMPG-α		0.9635	0.9625	0.9585	0.9605
	LMPG	$\lambda = 0.1$, $s = 0.05\lvert P\rvert$	**0.9708**	**0.9685**	**0.9685**	**0.9685**
	SVM	C = 1, d = 2	**0.9708**	**0.9685**	**0.9685**	**0.9685**
t-t-t	LMPG*		**1.0000**	**1.0000**	**1.0000**	**1.0000**
	LMPG-α		**1.0000**	**1.0000**	**1.0000**	**1.0000**
	LMPG	$\lambda = 0.01$, $s = 0.2\lvert P\rvert$	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	SVM	C =10, d = 5	**1.0000**	**1.0000**	**1.0000**	**1.0000**
mnist-49	LMPG*		0.9935	0.9935	0.9935	0.9935
	LMPG-α		0.9935	0.9935	0.9935	0.9935
	LMPG	$\lambda = 0.01$, $s = 0.2\lvert P\rvert$	**0.9938**	**0.9938**	**0.9938**	**0.9938**
	SVM	C = 10^3, d = 4	0.9935	0.9935	0.9935	0.9935
segment	LMPG*		0.9524	0.9561	0.9569	0.9560
	LMPG-α		0.9545	0.9600	0.9586	0.9587
	LMPG	$\lambda = 0.1$, $s = 0.15\lvert P\rvert$	**0.9654**	**0.9692**	**0.9682**	**0.9684**
	SVM	C = 10, d = 2	0.9632	0.9670	0.9670	0.9665
w8a	LMPG*		-	-	-	-
	LMPG-α		-	-	-	-
	LMPG	$\lambda = 0.01$, $s = 0.2\lvert P\rvert$	0.9853	**0.9340**	0.7947	0.8502
	SVM	C = 10, d = 2	**0.9861**	0.9147	**0.8308**	**0.8677**

Table 2. Performance of all proposed methods against SVM with polynomial kernel. For each dataset the best results are highlighted in **boldface**. Missing values (-) indicate the computation did not end in a reasonable amount of time.

$\{0, 0.01, 0.02, 0.1, 0.2\}$ and we considered as sample size 5%, 10%, 15% and 20% of the whole number of preferences.

The plots presented in Figure 1 and 2 describe how the value of the game changes according to the dimension of the sample size, and λ. The baseline (red line) describes the optimal value of the game obtained by LMPG*, the continuous curve is the value given by LMPG$-\alpha$, while the dashed one is the value obtained using LMPG.

Both figures exhibit the same pattern: the game values produced using $\lambda = 0$ are significantly worse than the one achieved by LMPG*. The best value for λ to obtain small values of the game seems to be 0.01. It is possible to observe that, with sufficient sample size and a small λ greater than 0, the approximated



(a) sample size = 5%

(b) sample size = 10%

(c) sample size = 15%

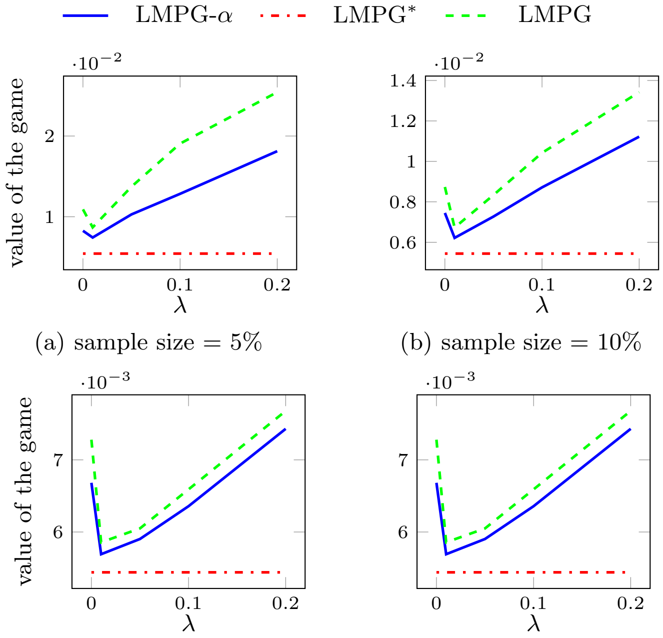
(d) sample size = 20%

Fig. 1. Value of the game on the `breast-cancer` dataset varying both the sample size and λ.

value obtained thanks to sampling and without the optimization of α is close enough to the optimal value and thus is able to perform well also in classification tasks. These findings reflect what was supposed theoretically in previous sections, especially about the values' magnitude ordering.

Figure 3 presents the accuracy results obtained by our algorithms using different sample sizes and λ. These charts show a pattern that follows the previous findings: generally speaking, a small λ produces the best results and again the sample size seems to be relevant to obtain good results, although in `segment` best results are obtained using the sample size equal to the 15% of the dataset.

The proposed strategies have been compared to soft SVM. SVMs have been validated using 5-fold validation: C has been validated in the set $\{1, 10, 10^2, 10^3\}$ and the degree of the homogeneous polynomial kernel in the range [1,5]. For our methods we used the best performing kernel (during validation) for SVM. Table 2 shows the comparison of the proposed technique with the λ and sample size that produce the best results against validated SVM. It is possible to observe that the proposed strategy performs better or as good as SVM in 4 out of 5 datasets (`tic-tac-toe`, `mnist-49`, `segment` and `breast-cancer`). The ranker produced by averaging over different strategies performs almost always (except on `tic-tac-toe`) better than the hypothesis obtained considering the optimal



(a) sample size = 5%

(b) sample size = 10%

(c) sample size = 15%

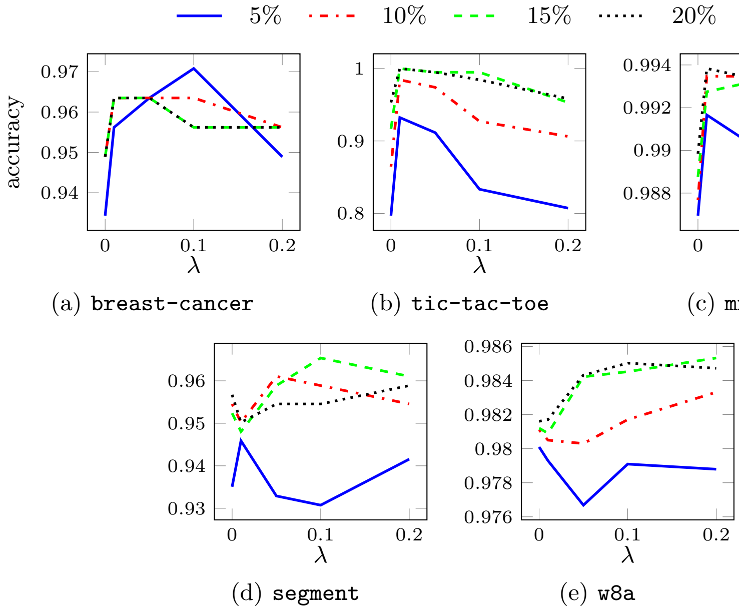
(d) sample size = 20%

Fig. 2. Value of the game on the `tic-tac-toe` dataset varying both the sample size and λ.

strategy distribution. This phenomenon can be explained by the fact that the optimal distribution corresponds to solving a hard margin problem, while the averaged one might represent a more soft solution. Note that this can be correlated with the low values for C obtained when validating SVM.

6 Conclusions

We proposed a principled game theoretical framework used for the multi-class classification task. We presented the mathematical formulation of a preference learning model able to solve the multi-class classification task as a single optimization problem. To reduce the complexity of the problem, we presented optimization strategies that exploit typical properties of reinforcement learning and solves reduced-size subproblems. In the experimental section, the proposed framework has exhibited state-of-the-art results. Among the future research paths we plan to explore, we aim to study the efficiency of the proposed algorithms. As already pointed out, using uniform sampling leads to a highly parallelizable version of the algorithm, yet results are not as good as those obtained by using adaptive sampling for preferences selection. One aspect of the theoretical framework that needs to be further developed is the study of the-



(a) `breast-cancer` (b) `tic-tac-toe` (c) `mnist-49`

(d) `segment` (e) `w8a`

Fig. 3. Accuracy of the proposed method varying the sample size (curves) and the value of λ (x axis) on (a) `breast-cancer`, (b) `tic-tac-toe`, (c) `mnist-49`, (d) `segment`, and (e) `w8a`.

oretical bounds limiting the differences in the value of the games when using different approaches. Empirical results suggest that these bounds can be strict provided mild assumptions, thus our technique can be easily applied to other domains.

References

1. Aiolli, F., Da San Martino, G., Sperduti, A.: A kernel method for the optimization of the margin distribution. In: Artificial Neural Networks - ICANN 2008. pp. 305–314 (2008)
2. Aiolli, F., Sperduti, A.: A preference optimization based unifying framework for supervised learning problems. In: Preference Learning (2010)
3. Bopardikar, S.D., Borri, A., Hespanha, J.P., Prandini, M., Benedetto, M.D.D.: Randomized sampling for large zero-sum games. In: 49th IEEE Conference on Decision and Control (CDC). pp. 7675–7680 (2010)
4. Bopardikar, S.D., Langbort, C.: Incremental approximate saddle-point computation in zero-sum matrix games. In: 53rd IEEE Conference on Decision and Control. pp. 1936–1941 (2014)
5. Brown, G.W.: Iterative solutions of games by fictitious play. In: Activity Analysis of Production and Allocation pp. 374–376 (1951)
6. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Willey & Sons (1973)

7. Freund, Y., Schapire, R.E.: Game theory, on-line prediction and boosting. In: COLT. pp. 325–332 (1996)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (Aug 1997)
9. Freund, Y., Schapire, R.E.: Adaptive game playing using multiplicative weights. Games and Economic Behavior **29**(1-2), 79–103 (1999)
10. Fürnkranz, J., Hüllermeier, E.: Preference Learning. Springer, 1st edn. (2010)
11. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification: A new approach to multiclass classification. In: Proceedings of the 13th International Conference on Algorithmic Learning Theory. pp. 365–379. ALT '02, Springer-Verlag, Berlin, Heidelberg (2002)
12. Hayashi, Y., Nakano, S.: Use of a recursive-rule extraction algorithm with j48graft to achieve highly accurate and concise rule extraction from a large breast cancer dataset. Informatics in Medicine Unlocked **1**, 9 – 16 (2015)
13. Ioannidis, S., Loiseau, P.: Linear regression as a non-cooperative game. In: Proceedings of the 9th International Conference on Web and Internet Economics - Volume 8289. pp. 277–290. WINE 2013 (2013)
14. von Neumann, J.: Zur theorie der gesellschaftsspiele **100**, 295–320 (1928)
15. Nilsson, N.J.: Learning machines: foundations of trainable pattern-classifying systems (1965)
16. Polato, M., Aiolli, F.: A game-theoretic framework for interpretable preference and feature learning. In: Artificial Neural Networks and Machine Learning – ICANN 2018. pp. 659–668. Springer International Publishing (2018)
17. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. In: Proceedings of the Fourteenth International Conference on Machine Learning. pp. 322–330. ICML '97 (1997)
18. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edn. (1998)
19. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the Twenty-first International Conference on Machine Learning. pp. 104–. ICML '04, ACM (2004)
20. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)