RAIRO-Oper. Res. 51 (2017) 329–341
DOI: 10.1051/ro/2016024

RAIRO Operations Research
www.rairo-ro.org

A NOTE ON SUPERVISED CLASSIFICATION AND NASH-EQUILIBRIUM PROBLEMS

Nicolas Couellan[1]

Abstract. In this note, we investigate connections between supervised classification and (Generalized) Nash equilibrium problems (NEP & GNEP). For the specific case of support vector machines (SVM), we exploit the geometric properties of class separation in the dual space to formulate a non-cooperative game. NEP and Generalized NEP formulations are proposed for both binary and multi-class SVM problems.

Mathematics Subject Classification. 91A80, 68T05, 68Q32.

Received October 15, 2014. Accepted March 10, 2016.

1. Introduction

Support vector machines applied to classification problems have been under active investigation for years now. Well understood theory as well as extensive experimentation have demonstrated their good learning capabilities [14]. The demand for high training speed SVM algorithms due to the increasing size of datasets is giving new research challenges to the optimization communities [15]. State of the art SVM formulations are based on soft margin expressions with a small set of applicable loss functions. Efficient algorithms have been developed working either in primal or dual variable space (ex: [15]). With no intend to compete with state of the art SVM training techniques, we are discussing the SVM problem from a different optimization angle: the point of view of non-cooperative games.

Machine learning applied to games where models are constructed to discriminate bad from good strategies has been a quite active research area [6]. However, application of game theory to supervised machine learning has not really been investigated. There has been recent work on applying game theory to unsupervised learning such as clustering [9]. However, to the best of our knowledge, only little work has been performed on supervised classification. As an example, there has been some research on applied matrix game to multi-class classification problems where multiple pairwise binary classifications are merged into a matrix game framework to design the best combination of classifiers [13].

It is natural or even intuitive to raise the question of connections between classification and non-cooperative games. Optimal class separation seems to be the result of a compromise situation between two (in the case of binary classification) distinct strategies. If classes are considered as players, one class objective is to lie on one

Keywords. Supervised classification, support vector machine, multi-class SVM, Nash equilibrium, generalized Nash equilibrium, game theory.

[1] Institut de Mathématiques de Toulouse, UMR 5219, Université de Toulouse, UPS IMT, 31062 Toulouse cedex 9, France.
nicolas.couellan@math.univ-toulouse.fr

Article published by EDP Sciences

side of the separating hyperplane while the other intends to lie on the exact other side and both aim to be as far as possible from the hyperplane to ensure generalization. This seems to be the setup of a non-cooperative game strategy. This phenomenon is even more intuitive when one is dealing with more than two classes (multi-class problems). However, as we will see, in the multi-class case, the player will now be defined as pair of classes instead of individual classes.

The parallel that we draw here has several objectives: first, we would like to discuss classification and separation issues from a different perspective. Geometrically, the maximum margin separation problem that arises in SVM has mainly been addressed in the primal space. Duality theory gives an alternative geometric interpretation that we propose to exploit. Furthermore, taking the game theory point of view provides new interpretations of critical classification issues. Additionally, in the presence of many classes to separate, traditional methods perform successive binary separations that can become computationally expensive. Using relaxation and regularization methods (such as, for example, the use of regularized Nikaido−Isoda function based methods) for solving the resulting GNEP, one could design algorithms that process all classes at once avoiding "one-against-one" or "one-against-all" expensive iterative procedures. Finally, looking at the classification paradigm as a multi-player game opens the door to distributed agent-based solving processes such as autonomous multi-agents systems (AMAS) that may help in large scale distributed classification contexts. These systems distribute complex optimization tasks to a collection of agents that individually optimize some utility function while sharing a common objective with the other agents [4,17]. These systems attempt to find an equilibrium state and it is known that their process is, by essence, a game solving process [12]. As we will see later, the data required to define each player strategy and utility only depends on a subset of the complete data. Therefore, the NEP and GNEP SVM formulations could be thought as a method to distribute the classification work among distributed agents in the context of large datasets. It may also help in preserving some data privacy between agents as one agent (player) only require the knowledge of its corresponding data subset.

The article is organized as follows: Section 2 recalls the binary classification problem, develops the geometric interpretation in the dual space and describes how it can be formulated as a Nash equilibrium problem (NEP). Along the same ideas, Section 3 explains the more general multi-class training problem, its expression as a generalized Nash equilibrium (GNEP) problem and discusses its properties. Section 4 gives some concluding remarks and briefly discusses perspectives.

2. Binary classification with support vector machines

2.1. Problem statement

Consider a set of training vectors $\{x_i \in \mathbb{R}^n, i = 1, \dots, L\}$ and its corresponding set of labels $\{y_i \in \{-1, 1\}, i = 1, \dots, L\}$, where L is the number of training points and n is the number of attributes of each training point.

Traditionally, the linear soft margin SVM training problem is expressed as follows (see for example [14] for further details on the construction of the problem):

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i \\ \text{s.t.} \quad & y_i(w^\top x_i + b) + \xi_i \geq 1, \quad \xi_i \geq 0,\ i = 1, \dots, L, \end{aligned} \tag{2.1}$$

where ξ_i is a slack variable associated to a penalty term in the objective with magnitude controlled by C, a problem specific parameter. The vector w is the normal vector to the separating hyperplane ($w^\top x + b = 0$) and b is its relative position to the origin.

Problem (2.1) maximizes the margin $\frac{2}{\|w\|}$ between the two separating hyperplanes $w^\top x_i + b = 1$ and $w^\top x_i + b = -1$. The use of slack variables ξ_i penalizes data points that would fall on the wrong side of the hyperplanes.

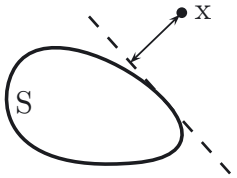

FIGURE 1. Distance from a point x to a convex set S.

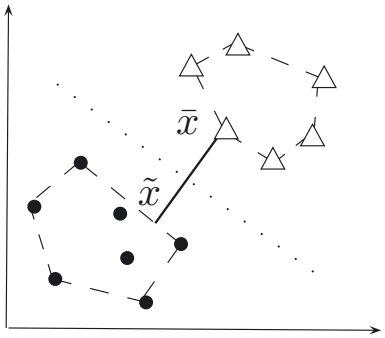

FIGURE 2. Minimizing the distance between the closest points of the convex hulls formed by the class points.

Dual formulations are sometimes preferred. From Lagrangian duality, the dual problem of problem (2.1) is:

$$\begin{aligned} \min_{\alpha} \quad & \tfrac{1}{2} \sum_{i,j=1}^{L} y_i y_j \alpha_i \alpha_j x_i^\top x_j - \sum_{i=1}^{L} \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{L} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, L. \end{aligned} \tag{2.2}$$

We now look at the above problem from a different perspective. Recall from duality theory (see [10]) that the minimum distance from a point to a convex set is the maximum of the distances from the point to the hyperplanes separating the point and the convex set (see also Fig. 1). Problem (2.2) can therefore be expressed as minimizing the distance between the convex hulls formed by the points from each class. Figure 2 illustrates the idea. The use of this geometric interpretation of duality in the context of SVM is also described in [2,3].

The problem of minimizing the (squared) distance between the two convex hulls can be formulated as follows:

$$\begin{aligned} \min_{\alpha} \ & \tfrac{1}{2} \|\tilde{x} - \bar{x}\|^2 \\ \text{s.t.} \ & \tilde{x} = \sum_{i \in S_{-1}} \alpha_i x_i, \quad \bar{x} = \sum_{i \in S_{+1}} \alpha_i x_i \\ & \sum_{i \in S_{-1}} \alpha_i = 1, \qquad \sum_{i \in S_{+1}} \alpha_i = 1 \\ & \alpha_i \ge 0 \qquad \text{for} \quad i = 1, \ldots, L \end{aligned} \tag{2.3}$$

with $S_{-1} = \{i / y_i = -1, i = 1, \ldots, L\}$ and $S_{+1} = \{i / y_i = +1, i = 1, \ldots, L\}$.

With the following Proposition 2.1, we state and prove that we can replace the standard dual formulation (2.2) by the above distance minimization problem.

Proposition 2.1. *If α is a solution of problem (2.3), then α is also a solution of (2.2).*

Proof. The solution α of problem (2.3) satisfies the following:

$$\tilde{x} = \sum_{i \in S_{-1}} \alpha_i x_i \qquad \bar{x} = \sum_{i \in S_{+1}} \alpha_i x_i$$

$$\sum_{i \in S_{-1}} \alpha_i = 1, \ \sum_{i \in S_{+1}} \alpha_i = 1, \quad \alpha_i \geq 0, \quad \text{for} \quad i = 1, \ldots, L.$$

Observe that $\sum_{i \in S_{-1}} \alpha_i x_i = -\sum_{i \in S_{-1}} \alpha_i y_i x_i$ and $\sum_{i \in S_{+1}} \alpha_i x_i = \sum_{i \in S_{+1}} \alpha_i y_i x_i$, therefore we have:

$$\begin{aligned} \frac{1}{2} \|\tilde{x} - \bar{x}\|^2 &= \frac{1}{2} \left\| \sum_{i=1}^{L} \alpha_i y_i x_i \right\|^2 \\ &= \frac{1}{2} \left(\sum_{i=1}^{L} \alpha_i y_i x_i \right)^\top \left(\sum_{i=1}^{L} \alpha_i y_i x_i \right) \\ &= \frac{1}{2} \sum_{i,j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^\top x_j \end{aligned}$$

and

$$\sum_{i=1}^{L} \alpha_i y_i = \sum_{i \in S_{-1}} \alpha_i y_i + \sum_{i \in S_{+1}} \alpha_i y_i = -\sum_{i \in S_{-1}} \alpha_i + \sum_{i \in S_{+1}} \alpha_i = 0.$$

The minimization problem (2.3) can then be reformulated as follows:

$$\begin{aligned} &\min_{\alpha} \ \tfrac{1}{2} \sum_{i,j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^\top x_j \\ &\text{s.t.} \sum_{i \in S_{-1}} \alpha_i = 1, \sum_{i \in S_{+1}} \alpha_i = 1, \quad \sum_{i=1}^{L} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \text{ for } \quad i = 1, \ldots, L. \end{aligned}$$

Since $\sum_{i \in S_{-1}} \alpha_i = 1$ and $\sum_{i \in S_{+1}} \alpha_i = 1$, we have $\sum_{i=1}^{L} \alpha_i = 2$, a constant that can be substracted to the objective function. The solution α is therefore solution of:

$$\begin{aligned} &\min_{\alpha} \ \tfrac{1}{2} \sum_{i,j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^\top x_j - \sum_{i=1}^{L} \alpha_i \\ &\text{s.t.} \sum_{i=1}^{L} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \qquad \text{for} \quad i = 1, \ldots, L. \end{aligned}$$

The constraints $\sum_{i \in S_{-1}} \alpha_i = 1$ and $\sum_{i \in S_{+1}} \alpha_i = 1$ in the above problem have been removed as they are satisfied since α is solution of problem (2.3). Finally, observe that the above problem is exactly the dual problem (2.2) when one consider the hard margin problem (no C penalty parameter) instead of the soft margin. Therefore α is solution of problem (2.2). □

The equivalence for the soft margin problem can also be shown if one uses the concept of *reduced convex hulls* limiting the convex combinations of points so that the class hulls do not intersect (see [3]). The idea is to limit the influence of points that can be seen as outliers and lead to non-separability (overlaps

of both convex hulls). This is achieved by adding a constraint $\alpha_i \leq D \, \forall i = 1, \ldots, L$ where D is a constant ($D < 1$) in problem (2.3). Reducing D sufficiently will ensure separability of the problem and this is actually equivalent to enlarging the margin by reducing the C parameter in problem (2.1). The introduction of an upper bound D on the α_i will not impact our developments in the next sections. Therefore, for simplicity and without loss of generality, we will only consider linearly separable datasets and hard margin formulations in the following.

As needed later, note that the separating hyperplane is located half way between the two convex hulls and perpendicular to the line segment defined by the optimal $w = \sum_{i=1}^{L} y_i \alpha_i x_i$. The distance from the origin to the intersection point with the line segment is given by $\frac{1}{2}|(\sum_{i \in S_{-1}} \alpha_i x_i - \sum_{i \in S_{+1}} \alpha_i x_i)^\top w|$ and the distance d between one class (convex hull) and the hyperplane is $\frac{1}{2} \|\tilde{x} - \bar{x}\| = \frac{1}{2} \|\sum_{i \in S_{-1}} \alpha_i x_i - \sum_{i \in S_{+1}} \alpha_i x_i\|$.

2.2. Reformulation as a Nash equilibrium problem

Consider two players: player 1 associated to class $+1$ and player -1 associated to class -1. Consider also that each player has the objective to minimize the closest distance between his class and the separating hyperplane. Clearly, if both players simultaneously interact with their own objective, they both will attempt to minimize the distance to the hyperplane in a non cooperative manner. We recall the following:

- The geometric distance from a point $z \in \mathbb{R}^n$ to a hyperplane defined by the equation $w^T x + b = 0$ is given by $\frac{|w^T z + b|}{\|w\|}$.
- In SVM, w defining the hyperplane can be expressed with the dual variable α *via* the following relation: $w = \sum_{i=1}^{L} y_i \alpha_i x_i$.

Without loss of generality and for simplicity, we omit the bias term b but its introduction would not change the principles of what follows.

Player 1 attempting to minimize the closest distance from class $+1$ to the separating hyperplane is therefore solving the following problem:

$$\begin{cases} \min\limits_{\alpha, w} \frac{|w^T \tilde{x}|}{\|w\|} \\ \text{s.t. } \tilde{x} = \sum\limits_{i \in S_{+1}} \alpha_i x_i, \ \sum\limits_{i \in S_{+1}} \alpha_i = 1, \ w = \sum\limits_{i=1}^{L} y_i \alpha_i x_i, \ \alpha_i \geq 0 \, \forall i = 1, \ldots, L \end{cases} \tag{2.4}$$

while Player -1 is solving the problem:

$$\begin{cases} \min\limits_{\alpha, w} \frac{|w^T \bar{x}|}{\|w\|} \\ \text{s.t. } \bar{x} = \sum\limits_{i \in S_{-1}} \alpha_i x_i, \ \sum\limits_{i \in S_{-1}} \alpha_i = 1, \ w = \sum\limits_{i=1}^{L} y_i \alpha_i x_i, \ \alpha_i \geq 0 \, \forall i = 1, \ldots, L. \end{cases} \tag{2.5}$$

Using the functional distance $|w^\top x|$ instead of the geometric distance $\frac{|w^\top x|}{\|w\|}$ and substituting the expression of w and $\tilde{x}$ in problem (2.4), we formulate an equivalent problem:

$$\begin{cases} \min\limits_{\alpha} \left| \left(\sum\limits_{i=1}^{L} y_i \alpha_i x_i \right)^\top \left(\sum\limits_{i \in S_{+1}} \alpha_i x_i \right) \right| \\ \text{s.t. } \sum\limits_{i \in S_{+1}} \alpha_i = 1, \ \alpha_i \geq 0 \quad \forall i = 1, \ldots, L. \end{cases}$$

Problem (2.5) could respectively be written in an equivalent form. To comply with standard notations from game theory [5], we reformulate both problems above into the general form:

NEP 2.2. Find $\bar{\alpha}^v \in S_v(\bar{\alpha}^{-v})$ for $v \in \{-1, 1\}$

where v denotes one player, $-v$ its adversary and $S_v(\bar{\alpha}^{-v})$ is the solution set of the following problem: $\min_{\alpha_v} \theta_v(\alpha^v, \alpha^{-v})$ s.t. $\alpha^v \in X_v(\alpha^v)$ and

$$X_v(\alpha^v) = \left\{ \alpha^v \ / \ \sum_{i \in I_v} \alpha_i^v = 1, \text{ and } \alpha_i^v \geq 0, \forall i \in I_v \right\}, I_v = \{i \ / \ y_i = v\}$$

$$\theta_v(\alpha^v, \alpha^{-v}) = \left| \left(\sum_{j \in I_{-v}} y_j \alpha_j^{-v} x_j + \sum_{i \in I_v} y_i \alpha_i^v x_i \right)^\top \left(\sum_{i \in I_v} \alpha_i^v x_i \right) \right|.$$

Next, we illustrate this problem formulation with simple examples.

Example 2.3. Consider 3 data points: $x_1 = (-1, 0)^\top$, $x_2 = (0, 1)^\top$, and $x_3 = (1, 0)^\top$ with corresponding labels $y_1 = +1$, $y_2 = -1$, and $y_3 = +1$. The problem of finding the separating hyperplane using the above GNEP formulation can be written as:

Player $+1$:

$$\theta_{+1}(\alpha) = \left| \begin{pmatrix} \alpha_3 - \alpha_1 \\ -\alpha_2 \end{pmatrix}^\top \begin{pmatrix} \alpha_3 - \alpha_1 \\ 0 \end{pmatrix} \right| = (\alpha_3 - \alpha_1)^2$$

$$X_{+1}(\alpha) = \{\alpha_1, \alpha_3 \ / \ \alpha_1 + \alpha_3 = 1 \text{ and } \alpha_i \geq 0, i = 1, 2, 3\}$$

Player -1:

$$\theta_{-1}(\alpha) = \left| \begin{pmatrix} \alpha_3 - \alpha_1 \\ -\alpha_2 \end{pmatrix}^\top \begin{pmatrix} 0 \\ \alpha_2 \end{pmatrix} \right| = \alpha_2^2$$

$$X_{-1}(\alpha) = \{\alpha_2 \ / \ \alpha_2 = 1 \text{ and } \alpha_i \geq 0, i = 1, 2, 3\}$$

It is easy to see that the solution of the above NEP is given by $\alpha = (1/2, 1, 1/2)^\top$ which leads to $w = \sum_{i=1}^{L} y_i \alpha_i x_i = (0, -1)^\top$ and $d = \frac{1}{2} \| \sum_{i \in S_{-1}} \alpha_i x_i - \sum_{i \in S_{+1}} \alpha_i x_i \| = \frac{1}{2}$. Figure 3 (*left*) illustrates the resulting separation.

Example 2.4. Consider 4 data points: $x_1 = (-1, 0)^\top$, $x_2 = (0, 1)^\top$, $x_3 = (1, 0)^\top$, and $x_4 = (0, -1)^\top$ with corresponding labels $y_1 = -1$, $y_2 = +1$, $y_3 = +1$, and $y_4 = -1$. The problem of finding the separating hyperplane using the above NEP formulation can be written as:

Player -1:

$$\theta_{-1}(\alpha) = \left| \begin{pmatrix} \alpha_1 + \alpha_3 \\ \alpha_2 + \alpha_4 \end{pmatrix}^\top \begin{pmatrix} -\alpha_1 \\ -\alpha_4 \end{pmatrix} \right| = \alpha_1(\alpha_1 + \alpha_3) + \alpha_4(\alpha_2 + \alpha_4)$$

$$X_{-1}(\alpha) = \{\alpha_1, \alpha_4 \ / \ \alpha_1 + \alpha_4 = 1 \text{ and } \alpha_i \geq 0, i = 1, 2, 3, 4\}$$

Player $+1$:

$$\theta_{+1}(\alpha) = \left| \begin{pmatrix} \alpha_1 + \alpha_3 \\ \alpha_2 + \alpha_4 \end{pmatrix}^\top \begin{pmatrix} \alpha_3 \\ \alpha_2 \end{pmatrix} \right| = \alpha_3(\alpha_1 + \alpha_3) + \alpha_2(\alpha_2 + \alpha_4)$$

$$X_{+1}(\alpha) = \{\alpha_2, \alpha_3 \ / \ \alpha_2 + \alpha_3 = 1 \text{ and } \alpha_i \geq 0, i = 1, 2, 3, 4\}$$

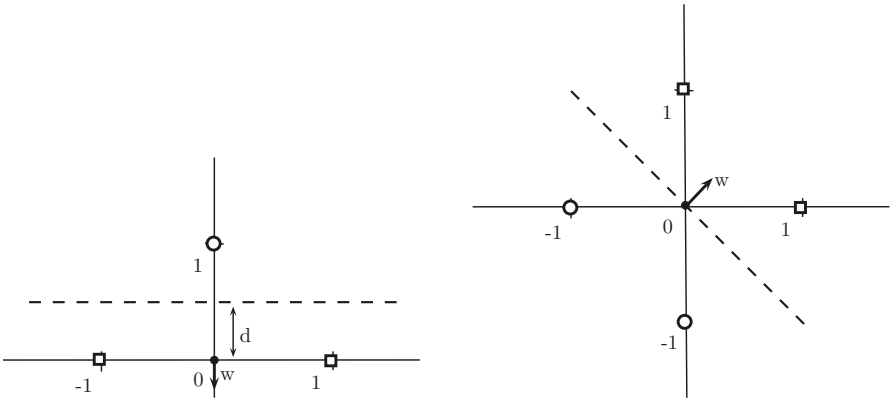

FIGURE 3. Examples of binary classification *via* GNEP.

This NEP has several solutions. One of these solutions is given by $\alpha = (1/2, 1/2, 1/2, 1/2)^\top$. The resulting vector w normal to the hyperplane is then $w = (1, 1)^\top$ and the distance d is $\frac{\sqrt{2}}{2}$. The separation for this example is shown on Figure 3 (*right*). Note that other solutions of this problem lead to the same value for w.

Observations/Interpretation:

- This game can be interpreted as if each player was trying to "pull" the hyperplane closer to himself. The utility function θ_v measures how close the hyperplane is to the player.
- *Case of class imbalance:* This situation arises when there is a majority class opposed to a minority class. This relates directly to the concept of fair/unfair game or biased/unbiased game in game theory. When the imbalance ratio is high, SVM may perfom poorly [8]. To avoid such situations, often under sampling or over sampling methods are used [16]. The idea is to add or remove some data points to balance the class cardinalities. Alternatively, it is possible to use cost sensitive learning methods that bias the estimation towards the minority class by applying a weighting scheme (i.e. bias the game in favor of the minority class player). From a game theory approach, all these techniques can be seen as empowering one of the player (the minority class) to artificially introduce "unfairness" in the game and restore balance between the classes.
- *The nonlinear case:* The above game formulation can also be extended to the nonlinear case. In this context, the game takes place in the feature space where linear separation is possible and for each player v, the utility function is replaced by $\hat{\theta}_v$ given by:

$$\hat{\theta}_v(\alpha^v, \alpha^{-v}) = \left| \left(\sum_{j \in I_{-v}} y_j \alpha_j^{-v} \varphi(x_j) + \sum_{i \in I_v} y_i \alpha_i^v \varphi(x_i) \right)^\top \left(\sum_{i \in I_v} \alpha_i^v \varphi(x_i) \right) \right|,$$

where φ is the map between the input space and the feature space. If one further develop the expression of $\hat{\theta}_v(\alpha^v, \alpha^{-v})$ using the distributivity of the scalar product, one would finally get an expression that only involves $\varphi(x_i)^\top \varphi(x_j)$ for the various values of i and j. This means that the *kernel trick*(see [14]) is applicable and the use of kernels is possible.

- *Situations with noisy data:* consider now the case where the input data x_i for $i = 1, \ldots, L$ is noisy and is therefore a random variable. The game takes place in the presence of uncertainty, meaning that there is a given probability to move from one strategy α^v to another strategy β^v. If the strategy sets were finite, this would exactly be the framework of well studied stochastic dynamic games (see for example [11]). Here, the strategy sets are not finite but in principles the idea remains the same and each player would like to minimize its expected utility $\mathbb{E}(\theta_v(\alpha^v, \alpha^{-v}))$.

3. Multi-class classification with support vector machines

3.1. Problem statement

Consider now a set of training vectors $\{x_i \in \mathbb{R}^n, i = 1, \ldots, L\}$ and its corresponding set of labels $\{y_i \in \{1, \ldots, M\}, i = 1, \ldots, L\}$, where L is again the number of training points, n the number of attributes of each training point and M the number of classes. Among the most common strategies and mathematical formulations for the multi-class problem, we can refer to the following methods [1]:

- *One-against-all:* M binary SVM models are constructed by taking the class k on one side and the other classes together as the opposite class ($k = 1, \ldots, M$). The resulting decision function will be of the form $y = \text{argmax}_{k=1,\ldots,M}\, w_k^\top x + b_k$ where (w_k, b_k) defines the optimal hyperplane computed by the kth binary model.
- *One-against-one:* $P = \frac{M(M-1)}{2}$ binary SVM models are constructed by taking each pair of classes in $\{1 \ldots, M\}$. The resulting decision function is obtained by a majority vote meaning that a point gets one vote for class k if the p-th pair of classes ($p = 1, \ldots, P$) assigns x to class k. The class with the highest total vote numbers will finally be assigned to x.
- *All-at-once:* The idea is to formulate the problem into one single optimization problem. It has the following expression:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\sum_{k=1}^{M} \|w_k\|^2 + C\sum_{i=1}^{L}\sum_{k\neq y_i} \xi_i^k \\ \text{subject to} \quad & w_{y_i}^\top x_i + b_{y_i} \geq w_k^\top x_i + b_k + 2 - \xi_i^k \\ & \xi_i^k \geq 0, i = 1, \ldots, L \\ & k \in \{1, \ldots, M\} \backslash y_i. \end{aligned} \tag{3.1}$$

The resulting decision function has the form: $y = \text{argmax}_{k=1,\ldots,M}\, w_k^\top x + b_k$.

Next, we propose a new formulation.

3.2. Reformulation as a generalized Nash equilibrium problem

Recall the formulation (2.3) of the problem of minimizing the distance between the closest points of the convex hulls formed by the classes. Consider now the generalization of this idea where one would like to minimize the pairwise distances between the various pairs of classes in the case of multi-class classification. If the problem has M classes, there are $P = C_M^2 = \frac{M(M-1)}{2}$ of such pairs and therefore P simultaneous optimization problems to be solved. As an illustration, consider the following example with 3 classes (class 1, class 2, class 3)

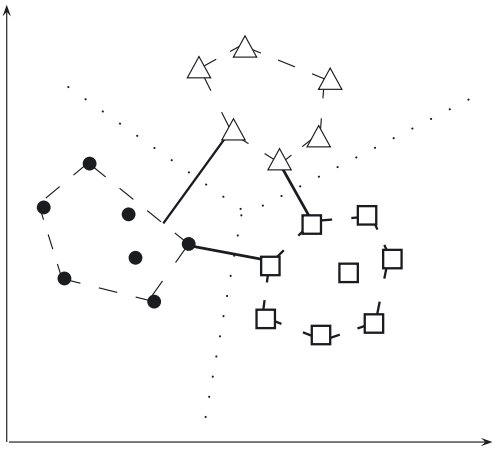

FIGURE 4. Minimizing pairwise distances between the closest points of convex hulls formed by the class points.

where 3 problems have to be solved (see also Fig. 4):

$$
\begin{cases}
\min\limits_{\alpha_i, i\in I_1\cup I_2} \frac{1}{2}\|\tilde{x}-\bar{x}\| \\
\text{s.t. } \tilde{x}=\sum\limits_{i\in I_1}\alpha_i x_i, \qquad \bar{x}=\sum\limits_{i\in I_2}\alpha_i x_i \\
\sum\limits_{i\in I_1}\alpha_i=1, \qquad \sum\limits_{i\in I_2}\alpha_i=1 \\
\alpha_i\geq 0 \qquad \text{for} \quad i=1,\ldots,L
\end{cases}
$$

$$
\begin{cases}
\min\limits_{\alpha_i, i\in I_2\cup I_3} \frac{1}{2}\|\bar{x}-\hat{x}\| \\
\text{s.t. } \bar{x}=\sum\limits_{i\in I_2}\alpha_i x_i, \qquad \hat{x}=\sum\limits_{i\in I_3}\alpha_i x_i \\
\sum\limits_{i\in I_2}\alpha_i=1, \qquad \sum\limits_{i\in I_3}\alpha_i=1 \\
\alpha_i\geq 0 \qquad \text{for} \quad i=1,\ldots,L
\end{cases}
$$

$$
\begin{cases}
\min\limits_{\alpha_i, i\in I_1\cup I_3} \frac{1}{2}\|\tilde{x}-\hat{x}\| \\
\text{s.t. } \tilde{x}=\sum\limits_{i\in I_1}\alpha_i x_i, \qquad \hat{x}=\sum\limits_{i\in I_3}\alpha_i x_i \\
\sum\limits_{i\in I_1}\alpha_i=1, \qquad \sum\limits_{i\in I_3}\alpha_i=1 \\
\alpha_i\geq 0 \qquad \text{for} \quad i=1,\ldots,L
\end{cases}
$$

where $I_k=\{i=1,\ldots,L : y_i=k\}$ for $k=1,2,3$.

Clearly, finding the best α at the intersection of all solution sets of these 3 problems is again a non cooperative game where each player is dealing with a pair of classes. The strategy of each player is to minimize the distance between the convex hulls formed by classes. This game can generally be expressed as follows:

GNEP 3.1. Find $\alpha^v \in S_v(\alpha^{-v})$ for $v\in\{1,\ldots,P\}$

where α^v denotes the strategy of player v (dealing with the pair of classes c and c' chosen among P pairs of classes) and α^{-v} the strategy of the other players and $S_v(\alpha^{-v})$ is the solution set of the following

problem: $\min_{\alpha_v} \theta_v(\alpha^v)$ s.t. $\alpha^v \in X_v(\alpha^{-v})$ with

$$X_v(\alpha^{-v}) = \left\{ \alpha^v \ : \begin{array}{l} \sum_{i\in I_c} \alpha_i^v = 1, \sum_{i\in I_{c'}} \alpha_i^v = 1, \\ \alpha_i^v, \alpha_j^{-v} \geq 0, \forall (i,j) \in (I_c \cup I_{c'}) \times \bigcup_{\substack{m=1 \\ m\neq c,c'}}^{M} I_m, \\ \forall k \in \{1,...,P\} \setminus \{v\} \left\{ \begin{array}{l} \alpha_i^v = \alpha_i^{-v_k} \ \forall i \in I_{c_k} \text{ if } c_k = c \\ \alpha_j^v = \alpha_j^{-v_k} \ \forall j \in I_{c'_k} \text{ if } c'_k = c' \end{array} \right. \end{array} \right\}$$

and

$$\theta_v(\alpha^v) = \frac{1}{2} \left\| \sum_{i\in I_c} \alpha_i^v x_i - \sum_{j\in I_{c'}} \alpha_j^v x_j \right\|,$$

where α^{-v_k} are the dual variables associated to the player involving the pair of classes (c_k, c'_k).

The above set $X_v(\alpha^{-v})$ expresses that the strategy α^v must generate points that belongs to the two convex hulls formed by the classes c and c' ($\sum_{i\in I_c} \alpha_i^v = 1, \sum_{i\in I_{c'}} \alpha_i^v = 1$ with $\alpha_i^v, \alpha_j^{-v} \geq 0$) and also the fact that each player associated to a pair of classes shares his classes with two other players ($\alpha_i^v = \alpha_i^{-v_k} \ \forall i \in I_{c_k}$ if $c_k = c$ or $\alpha_j^v = \alpha_j^{-v_k} \ \forall j \in I_{c'_k}$ if $c'_k = c'$). Its strategy α_v is therefore dependent on other players strategy α^{-v}.

The following example illustrates the formulation and its solution with a simple example:

Example 3.2. Consider the example on Figure 5 where we have 3 classes of points marked with $\square$ for class 1, $\triangle$ for class 2 and $\circ$ for class 3. Each class has only one data point. The data points are: $x_1 = (0,1)^\top$, $x_2 = (1,0)^\top$, and $x_3 = (0,-1)^\top$. The problem of finding the class separation using the above GNEP formulation can be written as:

Player 1:

$$\theta_1(\alpha^1) = \tfrac{1}{2} \left\| \begin{pmatrix} -\alpha_2^1 \\ \alpha_1^1 \end{pmatrix} \right\| = \tfrac{1}{2} \sqrt{(\alpha_1^1)^2 + (\alpha_2^1)^2}$$

$$X_1(\alpha^2, \alpha^3) = \left\{ \alpha^1 \ : \begin{array}{l} \alpha_1^1 = 1 \ , \ \alpha_2^1 = 1 \ , \ \alpha_1^1 = \alpha_1^3 \ , \ \alpha_2^1 = \alpha_2^2, \\ \alpha_i^1, \alpha_i^2, \alpha_i^3 \geq 0, i = 1,2,3 \end{array} \right\}$$

Player 2:

$$\theta_2(\alpha^2) = \tfrac{1}{2} \left\| \begin{pmatrix} \alpha_2^2 \\ -\alpha_3^2 \end{pmatrix} \right\| = \tfrac{1}{2} \sqrt{(\alpha_2^2)^2 + (\alpha_3^2)^2}$$

$$X_2(\alpha^1, \alpha^3) = \left\{ \alpha^2 \ : \begin{array}{l} \alpha_2^2 = 1 \ , \ \alpha_3^2 = 1 \ , \ \alpha_2^2 = \alpha_2^1 \ , \ \alpha_3^2 = \alpha_3^3, \\ \alpha_i^1, \alpha_i^2, \alpha_i^3 \geq 0, i = 1,2,3 \end{array} \right\}$$

Player 3:

$$\theta_3(\alpha^3) = \tfrac{1}{2} \left\| \begin{pmatrix} 0 \\ -\alpha_3^3 - \alpha_1^3 \end{pmatrix} \right\| = \tfrac{1}{2} \sqrt{(\alpha_3^1 + \alpha_3^3)^2}$$

$$X_3(\alpha^1, \alpha^2) = \left\{ \alpha^3 \ : \begin{array}{l} \alpha_3^3 = 1 \ , \ \alpha_1^3 = 1 \ , \ \alpha_3^3 = \alpha_3^2 \ , \ \alpha_1^3 = \alpha_1^1, \\ \alpha_i^1, \alpha_i^2, \alpha_i^3 \geq 0, i = 1,2,3 \end{array} \right\}$$

Each player tries to find the separating hyperplane between the pair he controls but his strategy is dependent on the moves of the other players. Here for example, Player 1 wants to minimize its utility $\theta_1(\alpha^1)$ where its

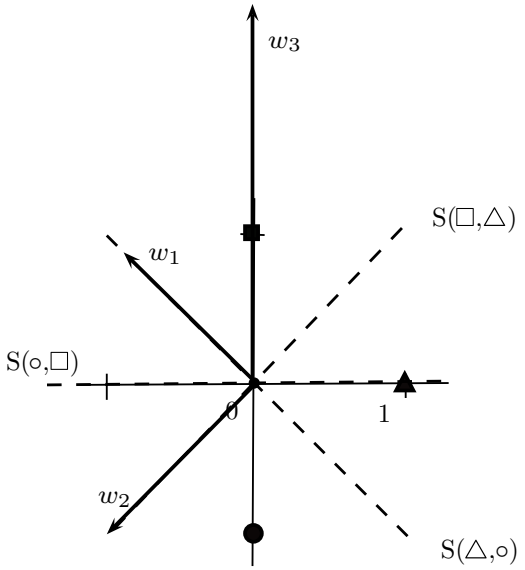

FIGURE 5. Example of multi-class classification *via* GNEP – $S(c, c')$ is the separating hyperplane between class c and class c' where $(c, c') \in \{(\square, \triangle), (\triangle, \circ), (\circ, \square)\}$.

strategy, α^1, must be in $X_1(\alpha^2, \alpha^3)$ defined by the strategy α^2 of Player 2 (by the fact that $\alpha_2^1 = \alpha_2^2$) and the strategy α^3 of Player 3 (by the fact that $\alpha_3^1 = \alpha_1^3$). Clearly, the solution of this GNEP is $\alpha^1 = (1,1)^\top$, $\alpha^2 = (1,1)^\top$ and $\alpha^3 = (1,1)^\top$ leading to $w_1 = (-1,1)^\top$, $w_2 = (-1,-1)^\top$ and $w_3 = (0,2)^\top$. Figure 5 illustrates the resulting separating hyperplanes.

Observations/Interpretation:

- The interpretation of this game differs from the previous section related to binary classification. Here, each game is a binary classification itself. Each player is trying to find the closest distance (and implicitly the maximum margin separating hyperplane) between a pair of classes. The overall game involves all pairs of classes with the ultimate goal to reach an equilibrium.
- *The jointly convexity property:* as illustrated in the following Proposition 3.3, it is easy to construct a jointly convex variant of problem (3.1). The jointly convex property is important as it ensures that Karush–Kuhn–Tucker conditions are sufficient optimality conditions for the GNEP. Theory of such problems is better understood and there are various types of algorithms available [5].

Proposition 3.3. *The variant of GNEP 3.1 using the objective* $\tilde{\theta_v}(\alpha^v) = \frac{1}{2}\left\|\sum_{i\in I_c}\alpha_i^v x_i - \sum_{j\in I_{c'}}\alpha_j^v x_j\right\|^2$ *is jointly convex.*

Proof. In order to prove Proposition (3.3), we have to verify the following:
(a) For each player, v, $\tilde{\theta_v}$ is a convex function.
(b) For each player, $X_v(\alpha^{-v})$ is a closed and convex set.
(c) For some closed convex set $X \subseteq \mathbb{R}^L$ and all players $v = 1, \ldots, P$, we have: $X_v(\alpha^{-v}) = \{\alpha^v \in \mathbb{R}^{|I_c \cup I_{c'}|} : (\alpha^v, \alpha^{-v}) \in X\}$

Clearly each objective function $\tilde{\theta_v}$ is a convex function. It is also easy to see that part (b) is satisfied by construction of each $X_v(\alpha^{-v})$. Each $X_v(\alpha^{-v})$ is defined by the following constraints:

$$\sum_{i \in I_c} \alpha_i^v = 1 \ , \ \sum_{i \in I_{c'}} \alpha_i^v = 1 \tag{3.2}$$

$$\alpha_i^v, \alpha_j^{-v} \geq 0 \ , \forall (i,j) \in (I_c \cup I_{c'}) \times \bigcup_{\substack{m=1 \\ m \neq c, c'}}^{M} I_m \tag{3.3}$$

$$\forall k \in \{1, ..., P\} \setminus \{v\} \begin{cases} \alpha_i^v = \alpha_i^{-v_k} \ \forall i \in I_{c_k} \text{ if } c_k = c \\ \alpha_j^v = \alpha_j^{-v_k} \ \forall j \in I_{c'_k} \text{ if } c'_k = c' \end{cases} \tag{3.4}$$

To verify part (c), we construct the following set X:

$$X = \left\{ \alpha^1, \ldots, \alpha^P \ : \begin{array}{c} \forall p \in \{1, \ldots, P\}, \\ \forall k \in \{1, \ldots, P\} \setminus \{p\} \begin{cases} \alpha_i^v = \alpha_i^{-v_k} \ \forall i \in I_{c_k} \text{ if } c_k = c_p \\ \alpha_j^v = \alpha_j^{-v_k} \ \forall j \in I_{c'_k} \text{ if } c'_k = c'_p \end{cases} \\ \alpha_i^p \geq 0, \alpha_j^p \geq 0, \forall (i,j) \in (I_{c_p} \times I_{c'_p}). \end{array} \right\}.$$

One can see that if α^v belongs to $X_v(\alpha^{-v})$, (α^v, α^{-v}) belongs to X, which satisfies part (c) of the proposition. □

- *The nonlinear case:* with multiple classes, one can easily see that the P players game can also take place in the feature space under the condition that the same kernel functions are used for each game. For each player the utility function $\hat{\theta}$ can then be expressed as:

$$\hat{\theta}_v(\alpha^v) = \frac{1}{2} \left\| \sum_{i \in I_c} \alpha_i^v \varphi(x_i) - \sum_{j \in I_{c'}} \alpha_j^v \varphi(x_j) \right\|,$$

  where φ is defined as before.
- *Class imbalance:* if one class dominates in number the other class in the pair, introducing artificially unfairness in the game taking place in this pair may also help in achieving better classification as seen in Section 2.2.
- *Multi-level game:* as already mentioned before, there are two levels of players. One game is taking place inside pairs (binary classification) while another game is taking place between the pairs. This framework is known as two-level game [7]. While it is interesting to make such observation, general forms of these types of games are very difficult to solve. Unless it can be shown that the very specific structure of the multi-level game that could be derived from GNEP 3.1 guarantees the existence of the multi-level Nash equilibrium (generalization of Nash-equilibrium concept to multi-level games), the one-level game formulation should be preferred (P players corresponding to the number of pairs of classes and prior computable knowledge of each player strategy is given).

4. Conclusions

We propose an alternative interpretation of the support vector machine classifier by making strong connections to game theory. We express binary and multi-class training problems as Nash and generalized Nash equilibrium problems involving several players. We define the utility of each player and discuss the game properties as well as the interpretation of important issues in machine learning such as class imbalance, presence of noise in the data or the nonlinear separation case. These interpretations are made in the context of games.

The initial objective of this note was to draw a parallel and unify some concepts of two mathematical fields that usually do not interplay in such manner. However, beyond the parallel that we propose, one may find use for the design of learning algorithms. The game theory formulation expresses naturally the problem as a distributed task among players. This structure could be exploited to decompose the training problem in the context of large scale distributed data. In formulation GNEP 3.1, the utility of each player k depends only on the pair of classes (c_k, c'_k) associated to player k and its strategy set depends only on the 2 classes shared with 2 other players. Therefore, the game that player k is playing is only based on a subset of the overall dataset. One can also see that, for the context of classification that requires some data privacy between players, some privacy is preserved as the data is not shared accross all players but only "neighbors" (the players that share a common class).

References

[1] S. Abe, Support Vector Machines for Pattern Classification. *Advances in Pattern Recognition*, 2nd edition. Springer, London, UK (2010).

[2] K. Bennett and E. Bredensteiner, Geometry in Learning, in *Geometry at Work*, edited by C. Gorini. Mathematical Association of America, Washington D.C. (2000) 132–145.

[3] K. Bennett and E. Bredensteiner, Duality and Geometry in SVMs, In *Proc. of 17th International Conference on Machine Learning*, edited by P. Langley. San Francisco (2000) 65–72.

[4] N. Couellan, S. Jan, T. Jorquera and J.-P. Georgé, Self Adaptive Support Vector Machine: A Multi-Agent Optimization Perspective. *Expert Syst. Appl.* **42** (2015) 4284–4298.

[5] F. Facchinei and C. Kanzow, Generalized Nash Equilibrium Problems. *Annals OR* **175** (2010) 177–211.

[6] J. Fürnkranz, Machine Learning in Games: A Survey, in *Machines that Learn to Play Games*. Nova Science Publishers (2001) 11–59.

[7] K. Hausken and R. Cressman, Formalization of Multi-level games. *Int. Game Theory Rev.* **6** (2004) 195–221.

[8] N. Japkowicz and S. Stephen, The class imbalance problem: A systematic study. *Intel. Data Anal.* **6** (2002) 429–449.

[9] G. Koltsidas and F.-N. Pavlidou, A Game Theoretical Approach to Clustering of Ad-Hoc and Sensor Networks. *Telecomm. Systems* **47** (2011) 81–93.

[10] D.G. Luenberger, *Optimization by Vector Space Methods*, 1st edition. John Wiley & Sons, Inc., New York, USA (1997)

[11] A. Neyman and S. Sorin (eds.), Stochastic Games and Applications, *Proc. of the NATO Advanced Study Institute, Stony Brook, New York, USA, 1999, Series: Nato Science Series C: (closed)*, Vol. 570 (2003)

[12] S. Parsons and M. Wooldridge, Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems* **5** (2002) 243–254.

[13] M. Petrovskiy, A Game Theory Approach to Pairwise Classification with Support Vector Machines. *Proc. of the 2004 International Conference on Machine Learning and Application. ICMLAs*, IEEE Computer Society (2004) 115–122.

[14] B. Schölkopf and A. Smola, *Learning with Kernels*. MIT, Cambridge (2002).

[15] S. Sra, S. Nowozin and S.J. Wright, *Optimization for Machine Learning*. MIT Press, Cambridge (2011).

[16] G.M. Weiss, Mining with Rarity: A Unifying Framework. *ACM SIGKDD Explorations Newsletter* **6** (2004) 7–19.

[17] G. Weiss, A modern Approach to Distributed Artifical Intelligence. *Intelligent Robotics & Autonomous Agents Series*, MIT Press, Cambridge (2000).

[18] J. Weston and C. Watkins, Support Vector Machines for Multi-Class Pattern Recognition. *Proc. of ESANN'1999, Bruges, Belgium* (1999) 219–224.