Support Vector Machine As Graph Theory Problems

1st William Brendel
Snap Research
Snap Inc.
Los Angeles, USA
william.brendel@snap.com

2nd Luis Marujo
Snap Research
Snap Inc.
Los Angeles, USA
luis.marujo@snap.com

***Abstract*— Quadratic programming (QP) problem reformulation has been a research problem for nearly two decades, but is seldom linked to Graph Theory. In fact, typical reformulations convexify a non-convex QP problem. This is accomplished by making the objective function differentiable, optimizing in the continuous domain while ensuring the final solution is binary, or adding regularizers and Lagrangian coefficients to optimize the dual problem. In this research, we demonstrate that QP problems can also be reformulated using the same mechanism as P/NP problem reduction, overcoming speed and memory footprint limitations from other type of reformulation. We use SVM to make the demonstration. In the demonstration, we show that SVM is comparable to a soft weighted edge maximum independent set problem where the amount of support vectors per class is balanced. As a result, SVM can also be reformulated as a maximum clique problem with the same class balancing constraint. After transforming the sequential minimal optimization (SMO) algorithm to our new maximum clique formulation, we demonstrate that such reformulation leads to improved training performance, reaching 36 times faster training time, and a sparser solution for less than one percent accuracy degradation in some datasets.**

***Index Terms*—Quadratic Optimization, Classification algorithms, Support Vector Machines, Graph Theory, Independent Set, Maximum Weighted Clique.**

I. Introduction

In the fields of Machine Learning, Natural Language Processing, Computer Graphics, and Computer Vision, there are multiple problems that can be solved using a quadratic programming (QP) solution. They are usually are formulated as graph theory problems, where typical objective functions to be optimized contain unary potentials related to nodes and binary potentials related to edges. There is a long 50-years history where graph theory has been at the core of an ocean of computer science applications [1], [2]. For instance, image segmentation has been formulated as minimum cut [3], [4], maximum weight independent set [5], maximum weight clique [6], [7] and minimum spanning tree [8] problems [9]. Multi-object tracking has been modeled as maximum weight independent set [10] and generalized minimum and maximum clique [11], [12] problems. Nevertheless, not all QP problems are directly related to graph theory as demonstrated by [13]–[17]. Since graph problems can be reduced to one another, QP problems can also be reformulated into problems that can be solved more efficiently. Several cut problems are reformulated as spectral clustering problems that can be optimized via weighted kernel k-means algorithms [4], [18], achieving real-time computation performances. In the work of Tsang et al. [19], SVM and Support Vector Clustering (SVC) are formulated as minimum enclosing ball problems, obtaining approximate optimal solutions in linear time, with a space complexity independent of the problem size. This work focuses on how SVM formulation [20] can be viewed as an independent set problem [19], [21], and thus be reduced to other graph problems.

More precisely, we connect the SVM dual QP formulation Eqn. (1) to a maximum independent set formulation, which gives a new interpretation on the support vector selection process. Then, the same way we reduce an independent set problem to a maximum clique problem by taking the complement of a graph, we reformulate the SVM dual QP formulation as a dominant set QP [6], [7], the latter being used to define the maximum clique (MC[1]) problem [21], [22]. We show that our MC formulation involves Mercer distance kernels, instead of Mercer similarity kernels, and demonstrate how to construct such kernels, leading to new families of kernels. Finally, we show that our MC formulation has computational advantages while preserving comparable accuracy compared to the standard LIBSVM implementation of the SVM dual QP formulation.

The remainder of this paper is organized as follows: Section II gives a review of the SVM Dual Formulation and SMO, Section III introduces the Problem and how we reformulated it, Section IV includes the experiments and result, Section V presents the conclusions and future work.

II. SVM Dual Formulation And SMO

SVM aims at learning boundaries between feature vectors $\{\mathbf{x}^k\}_{k=1}^n$ of different classes $\{\mathbf{y}_k\}_{k=1}^n$. When classes are binary, SVM dual formulation has the following form [20], [23]:

$$\begin{aligned} \boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in[0,\,\mathrm{C}]^n}{\arg\max} \quad & \mathcal{F}(\boldsymbol{\alpha}) : \mathbf{1}^\intercal\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\intercal\mathbf{H}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{y}^\intercal\boldsymbol{\alpha} = 0 \quad \text{with} \quad \mathbf{y} \in \{-1,\,1\}^n \\ & \mathbf{H}_{ij} = \mathbf{y}_i\mathbf{y}_j K(\mathbf{x}^i,\,\mathbf{x}^j) \;\text{ and }\; \mathbf{x}^k \in I\!R^d \end{aligned} \tag{1}$$

where $\boldsymbol{\alpha}_k > 0$ means that $\mathbf{x}^k$ is a support vector for class $\mathbf{y}_k$. i.e. the boundary is defined as a linear combination of $\{\mathbf{x}^k \,|\, \boldsymbol{\alpha}_k > 0\}$ in a vector space where the dot product is induced by the kernel function K. SMO was originally designed

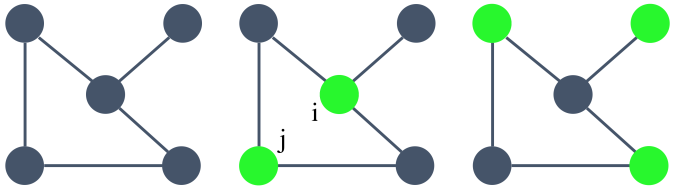
[1] MC = Maximum Clique, not to be confused with Monte Carlo.

to train a support vector machine that requires the solution of the very large SVM dual QP optimization problem [24], defined above. SMO belongs to the family of SQP algorithms [25]. It breaks a large QP problem into a series of smaller QP problems, each of which optimizes a quadratic model of the objective subject to a linearization of the constraints. The method is equivalent to applying Newton's method to the Karush-Kuhn-Tucker (KKT) conditions of the QP problem. In SMO, the small QP problems involve only two variables. These small QP problems are solved analytically, thus avoiding the use of a time-consuming numerical QP optimization at each iteration. The amount of memory required for SMO is linear in the training set size n, which allows SMO to handle very large input sets.

III. Problem Setup and Reformulation

Typical mathematical formulations of graph theory problems involve optimizing objective functions that contain unary potentials related to nodes, and binary potentials related to edges. Unary potentials are embedded in the form of a vector, and binary potentials in the form of an adjacency matrix. The latter is symmetric if the graph is undirected and semi-definite if the graph theory problem to be solved is P-complete. In the following, we outline how SVM can be interpreted as a special maximum independent set (MIS), and how it can be reformulated as a balanced maximum clique (MC). The two steps reformulation (SVM → MIS → MC) offers several advantages: it provides another interpretation on the SVM maximum margin formulation, which helps in designing new algorithms, and it also gives us the opportunity to take advantage of 50 years of research and algorithm development of about the MC problem, like the replicator dynamics approach from [26]. The MC formulation is particularly interesting as it provides a natural control over the sparsity of the solution in terms of number of support vectors.

A. SVM As a Balanced Maximum Independent Set Problem



(a) Graph $\mathcal{G}$ (b) An IS (c) The MIS

Fig. 1. A 5-vertices graph $\mathcal{G}$, with an independent set of size 2, and the maximum independent set of size 3. Vertices in the (maximum) independent set are displayed in green. Best viewed in color.

In graph theory, an independent set (IS) is a set of vertices in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, no two of which are adjacent. That is, it is a set $\mathcal{S} \subset \mathcal{V}$ of vertices such that for every two vertices i, j in $\mathcal{S}$, there is no edge connecting the two: $\forall i, j \in \mathcal{S}, (i, j) \notin \mathcal{E}$. Equivalently, each edge in the graph has, at most, one endpoint in $\mathcal{S}$. The size $|\mathcal{S}|$ of an independent set is the number of vertices it contains. A maximum independent set (MIS) is an independent set of largest possible size for a given graph $\mathcal{G}$. When each vertex is assigned a weight, the maximum weighted independent set (MWIS) is the subset of vertices whose weights sum to the maximum possible value without any two vertices being adjacent to one another. Both MIS and MWIS problem are NP-hard [27]. We illustrate the concept of IS and MIS in Fig. (1). The MIS problem is mathematically formulated as follows. Let's assume that a graph $\mathcal{G}$ contains n vertices. We define $\boldsymbol{\alpha} \in \{0, 1\}^n$ as the support[2] of $\mathcal{S}$, i.e. $\forall i \in \mathcal{V}, i \in \mathcal{S} \Leftrightarrow \boldsymbol{\alpha}_i = 1$, otherwise $\boldsymbol{\alpha}_i = 0$. We represent the edge set $\mathcal{E}$ by the adjacency matrix $\mathbf{A}$, i.e. $(i, j) \in \mathcal{E} \Leftrightarrow \mathbf{A}ij = 1$. [21] demonstrates that $\frac{1}{2}\boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha}$ is the number of edges between vertices in $\mathcal{S}$ and that $\mathbf{1}^\intercal\boldsymbol{\alpha} = |\mathcal{S}|$. Hence, $\mathcal{S}$ is an independent set if and only if $\boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha} = 0$, and the maximum independent set problem can be formulated as:

$$\boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in\{0,1\}^n}{\arg\max}\ \mathbf{1}^\intercal\boldsymbol{\alpha} \quad \text{s.t. } \boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha} = 0 \tag{2}$$

In addition, [21] shows that the latter equation can be reformulated as:

$$\boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in\{0,1\}^n}{\arg\max}\ \mathbf{1}^\intercal\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha} \tag{3}$$

and that we can loosen the binary constraint $\boldsymbol{\alpha} \in \{0, 1\}^n$ to the continuous domain $\boldsymbol{\alpha} \in [0, 1]^n$. We can further add a regularization diagonal matrix $\mathbf{D}$: $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{D}$, such that the final solution entries $\boldsymbol{\alpha}_i^\star$ are pushed to the domain boundaries [22], leading to the final MIS problem formulation:

$$\boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in[0,1]^n}{\arg\max}\ \mathbf{1}^\intercal\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\intercal(\mathbf{A} - \mathbf{D})\boldsymbol{\alpha} \tag{4}$$

We can clearly see that Eqn. (1) resembles the independent set QP formulation Eqn. (3), with three differences. First, $\boldsymbol{\alpha} \in [0, 1]^n$ instead of $\boldsymbol{\alpha} \in [0, \mathrm{C}]^n$ in Eqn. (1), second $\mathbf{H} \in \mathbb{R}^{n\times n}$ whereas $\mathbf{A} \in \{0, 1\}^{n\times n}$, and third Eqn. (1) has an additional constraint $\mathbf{y}^\intercal\boldsymbol{\alpha} = 0$. Nevertheless, we can reformulate both Eqn. (1) and Eqn. (3) to establish clearer correspondences.

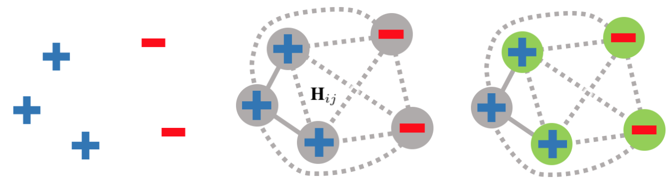
In Eqn. (1), we remap $\boldsymbol{\alpha} \leftarrow \frac{1}{\mathrm{C}}\boldsymbol{\alpha}$ so that $\boldsymbol{\alpha} \in [0, 1]^n$. This is equivalent to multiplying the kernel K by C. Without a loss of generality, we can further normalize K so that $\mathbf{H} \in [-1, 1]^{n\times n}$. Note that the new normalized kernel is still a valid SVM Mercer kernel, since the normalization is linear, hence it preserves the convexity of $\mathbf{H}$.

In Eqn. (3), we relax $\mathbf{A}_{ij} \in [0, 1]$ to represent an edge probability rather than an hard binary edge connectivity. Then we remap $\mathbf{A} \leftarrow 2\mathbf{A} - \mathbf{1}^\intercal\mathbf{1}$ so that $\mathbf{A}_{ij} \in [-1, 1]$. This can be interpreted as mapping the edge probabilities to a correlation metric. Moreover, this does not affect the solution of the MIS formulation as it just adds an additional maximizer $(\mathbf{1}^\intercal\mathbf{x})^2$ to the objective function in Eqn. (2) and (3). Indeed, the function $f\colon z \to z + z^2$ is strictly monotonically increasing on $\mathbb{R}_+$, hence maximizing $f(\mathbf{1}^\intercal\mathbf{x})$ is the same as maximizing $\mathbf{1}^\intercal\mathbf{x}$. In the next section we also see that this additional maximizer

[2] Here we use the symbol $\boldsymbol{\alpha}$ instead of $\mathbf{x}$ to draw the parallel with Eqn. (1).

$(\mathbf{1}^\intercal\mathbf{x})^2$ vanishes in the MC formulation Eqn. (5) due to the MC sparsity constraint.

As illustrated Fig. (2), we see now that support vectors can be viewed as nodes in $\mathcal{G}$, connected to each other with a soft correlation weight $\mathbf{H}_{ij}$ on edges (instead of an edge probability), where $\mathbf{H}_{ij} > 0$ means that support vectors i and j are likely connected on the graph, and $\mathbf{H}_{ij} < 0$ means that they are likely disconnected on the graph. $\mathbf{H}_{ij} = 0$ would be equivalent to having an edge probability of 0.5.



(a) 2-classes dataset (b) Graph from $\mathbf{H}$ (c) The MIS

Fig. 2. Independent set formulation of SVM. Edges with high $\mathbf{H}_{ij}$ weight are plain, edges with low $\mathbf{H}_{ij}$ weight are dashed. Green nodes represent the support vectors as a MIS. Best viewed in color.

Finally, the additional constraint $\mathbf{y}^\intercal\boldsymbol{\alpha} = 0$ imposes the solution to be balanced: the amount of support vectors for each class should be the same.

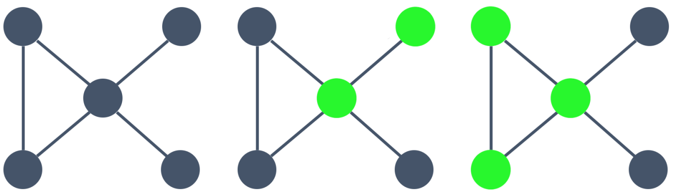
B. SVM Interpretation

Relating SVM to a maximum independent set problem induces the following interpretation: SVM selects support vectors (i) that are either not from the same class and highly correlated, or that are from the same class and as dissimilar as possible, (ii) such that the amount of support vectors for each class is balanced. It means that SVM aims to maximize support vectors with small margin (as the original max-min SVM formulation [28]) between classes that summarize locally the decision boundary intra classes. The next section further emphasizes even more the notion of boundary summarization with the additional sparsity constraint on the solution when Eqn. (1) is reformulated as a maximum weighted clique problem.

Tsang et al. [19] formulates SVM and SVC as approximations to the minimum enclosing ball problem, allowing them to train support vectors on very large datasets. Their algorithm iteratively alternates between refining the center of the enclosing ball and finding the next support vector as far away from it as possible. The center is defined implicitly: only the similarity from a point to the center is defined explicitly as the linear combination of the support vector kernels. Our interpretation fully supports their approach as we showed that support vectors are far away from each other intra-class and as close as possible to each other extra-class, which is the essence of the max-margin formulation. It also means that we can extend the strategy of [19] to initialize the solution by greedily preselecting a group of pairs of data points as initial support vectors such that points in each pair are from different classes, and such that the distance intra-class is maximized and the distance extra-class is minimized. This strategy helps reduce the number of iterations and thus the training time as demonstrated by Table II. Next we show that if SVM is viewed as an MIS problem, it can also be reduced to a MC problem.

C. SVM As a Balanced Maximum Clique Problem

Given a graph $\widetilde{\mathcal{G}}(\mathcal{V}, \mathcal{E})$ with n vertices, a clique is a subset of vertices $\mathcal{S} \subset \mathcal{V}$, all connected to each other, i.e. $\forall i, j \in \mathcal{V}, (i, j) \in \mathcal{E}$. The size $|\mathcal{S}|$ of a clique is the number of vertices it contains. A maximum clique (MC) is a clique of largest possible size for a given graph $\mathcal{G}$. When each vertex is assigned a weight, the maximum weighted clique (MWC) is the subset of vertices all adjacent to each other and whose weights sum to the maximum possible value. Both MC and MWC problem are NP-hard [27], as well as enumerating all possible clique in a graph. We illustrate the concept of clique and MC in Fig. (3).



(a) Graph (b) A clique (c) The MC

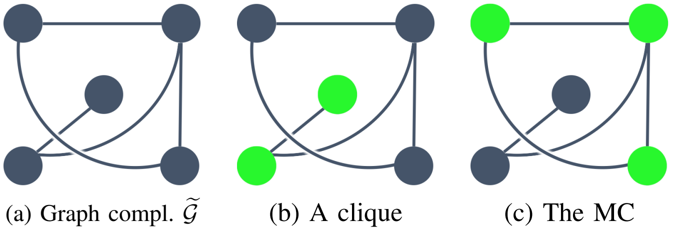
Fig. 3. A 5-vertices graph, with a clique of size 2, and the maximum clique of size 3. Vertices in the (maximum) clique are displayed in green. Best viewed in color.

In [29], Motzkin and Straus showed that the maximum clique problem is equivalent to the following continuous quadratic program:

$$\boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in I\!R^n_+}{\arg\max}\ \frac{1}{2}\,\boldsymbol{\alpha}^\intercal\widetilde{\mathbf{A}}\boldsymbol{\alpha} \quad \text{s.t.}\ \ \mathbf{1}^\intercal\boldsymbol{\alpha} = 1 \tag{5}$$

where $\widetilde{\mathbf{A}}$ is the adjacency matrix for the graph $\widetilde{\mathcal{G}}$, and $\boldsymbol{\alpha} \in I\!R^n_+$ is the support of $\mathcal{S}$, i.e. $\forall i \in \mathcal{V}, i \in \mathcal{S} \Leftrightarrow \boldsymbol{\alpha}_i > 0$. More precisely, there exists a solution $\boldsymbol{\alpha}^\star$ to Eqn. (5) such that (i) every nonzero component of $\boldsymbol{\alpha}^\star$ is equal to $\frac{1}{k}$, where k is the maximum cardinality of a clique in $\widetilde{\mathcal{G}}$; and (ii) the set $\mathcal{S} = \text{supp}(\boldsymbol{\alpha}^\star)$ is a clique of size k. Typically $\widetilde{\mathbf{A}}_{ij} \in \{0, 1\}$, but having $\widetilde{\mathbf{A}}_{ij} \in \{-1, 1\}$ does not change the formulation, as the transformation $\widetilde{\mathbf{A}} \leftarrow 2\widetilde{\mathbf{A}} - \mathbf{1}^\intercal\mathbf{1}$ introduces an additional penalty term $-(\mathbf{1}^\intercal\boldsymbol{\alpha})^2$ that vanishes with the constraint $\mathbf{1}^\intercal\boldsymbol{\alpha} = 1$.

The MC and MIS problems are highly correlated: the solutions of the MIS problem on a graph $\mathcal{G}$ are the same as the solutions of the MC on the complement $\widetilde{\mathcal{G}}$ of the graph $\mathcal{G}$, as illustrated Fig. (4). Similarly, we go from Eqn. (3) to Eqn. (5) by taking the complement $\widetilde{\mathbf{A}}$ of the adjacency matrix $\mathbf{A}$, i.e. $\widetilde{\mathbf{A}} \leftarrow \mathbf{1}\mathbf{1}^\intercal - \mathbf{A}$. Indeed, the objective function of Eqn. (3) becomes $\mathbf{1}^\intercal\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\intercal\widetilde{\mathbf{A}}\boldsymbol{\alpha} = \mathbf{1}^\intercal\boldsymbol{\alpha}(1 - \frac{1}{2}\mathbf{1}^\intercal\boldsymbol{\alpha}) + \frac{1}{2}\boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha}$. Since a maximizer $\boldsymbol{\alpha}^\star$ of $\boldsymbol{\alpha}^\intercal\mathbf{A}\boldsymbol{\alpha}$ is up to a constant factor, and since the function $f\colon z \to z(1 - \frac{1}{2}z)$ has a strict maximum at $z^\star = 1$, we can eliminate $f(\mathbf{1}^\intercal\boldsymbol{\alpha}) = \mathbf{1}^\intercal\boldsymbol{\alpha}(1 - \frac{1}{2}\mathbf{1}^\intercal\boldsymbol{\alpha})$ from the objective function, add the constraint $\mathbf{1}^\intercal\boldsymbol{\alpha} = 1$ and relax $\boldsymbol{\alpha}$ to the continuous domain, resulting the formulation Eqn. (5).



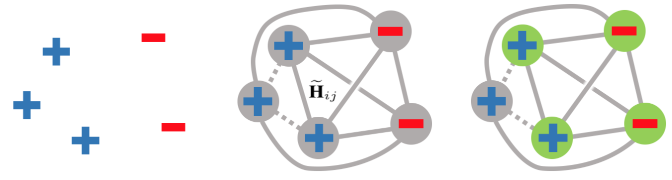
(a) Graph compl. $\widetilde{\mathcal{G}}$ (b) A clique (c) The MC

Fig. 4. Graph complement $\widetilde{\mathcal{G}}$ to the graph $\mathcal{G}$ in Fig. (1), with a clique of size 2, and the maximum clique of size 3. The vertices in $\widetilde{\mathcal{G}}$'s MC are the same as the vertices in $\mathcal{G}$'s MIS. Vertices in the (maximum) clique are displayed in green. Best viewed in color.

The dual SVM formulation is similar to the MIS formulation [21], with the additional constraint $\mathbf{y}^\intercal\boldsymbol{\alpha} = 0$ and where the graph edge weights have soft values (i.e. $\mathbf{H}_{ij}$ in Eqn. (1)). Since we can switch from the MIS problem to MC problem by taking the complement of the graph, we can then reformulate the SVM dual formulation in Eqn. (1) as:

$$\begin{aligned} \boldsymbol{\alpha}^\star \leftarrow \underset{\boldsymbol{\alpha}\in\mathbb{R}^n_+}{\arg\max}\quad & \mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\intercal\widetilde{\mathbf{H}}\boldsymbol{\alpha} \\ \text{s.t.}\quad & \mathbf{y}^\intercal\boldsymbol{\alpha}=0 \text{ and } \mathbf{1}^\intercal\boldsymbol{\alpha}=\nu \\ \text{with}\quad & \widetilde{\mathbf{H}}_{ij} = \mathbf{y}_i\mathbf{y}_j\widetilde{K}(\mathbf{x}^i,\mathbf{x}^j)\\ \text{and}\quad & \mathbf{x}^k\in\mathbb{R}^d, \quad \mathbf{y}\in\{\text{-}1,1\}^n \end{aligned} \tag{6}$$

where $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j)$ is now a Mercer kernel representing the distance between the two vectors $\mathbf{x}^i$ and $\mathbf{x}^j$ instead of their similarity. We illustrate our new formulation in Fig. (5).



(a) 2-classes dataset (b) Graph from $\widetilde{\mathbf{H}}$ (c) The MC

Fig. 5. Maximum clique formulation of SVM. Edges with high $\widetilde{\mathbf{H}}_{ij}$ weight are plain, edges with low $\widetilde{\mathbf{H}}_{ij}$ weight are dashed. Green nodes represent the support vectors as a MC. Best viewed in color.

We generalize the constraint $\mathbf{1}^\intercal\boldsymbol{\alpha} = 1$ to $\mathbf{1}^\intercal\boldsymbol{\alpha} = \nu$ to have an additional control on the ℓ_1 sparsity of the solution. Note that like in Eqn. (4), [7] shows that we can add a regularization diagonal matrix $\mathbf{D}$ to $\widetilde{\mathbf{H}}$ to guide the solution. For the MC formulation, various choice of $\mathbf{D}$ have been proposed by [6], [7], but we haven't noticed significant improvements in our experiments by adding $\mathbf{D}$. Hence for clarity we'll just focus on using $\widetilde{\mathbf{H}}$ only. Next, we demonstrate how to construct a proper Mercer distance kernel.

D. Mercer Distance Kernels

First of all, we can see that if $K(\mathbf{x}^i, \mathbf{x}^j) : \mathbb{R}^n \to \mathbb{R}$ is a similarity measure, $\exists\,\rho,\, \upsilon \in \mathbb{R}$ such that we can build a distance metric $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) : \mathbb{R}^n \to \mathbb{R}$ from a similarity measure using one of the following transformations:

$$\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{\rho}{\upsilon + K(\mathbf{x}^i, \mathbf{x}^j)} \quad \text{or} \quad \widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \upsilon - K(\mathbf{x}^i, \mathbf{x}^j) \tag{7}$$

Note that other transformation functions like $x \to e^{-\upsilon x^\rho}$ can be used as well. Proof and new kernel construction can be found in [30]. Interestingly enough, we can build a similarity measure from a distance metric the same way.

Lemma 1. *$\exists\,\rho,\, \upsilon \in \mathbb{R}$ such that Eqn. (7) always produces a valid Mercer kernel.*
Proof sketch. We base the proof on the Gershgorin circle theorem. We can find υ and ρ such that:

$$\text{min eigenvalue} \geq \min_i \left(\widetilde{K}(\mathbf{x}^i, \mathbf{x}^i) - \sum_{j\neq i} \widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) \right) \geq 0 \tag{8}$$

Since the Gershgorin bound is very loose, solving Eqn. (8) will produce a final matrix where $|\widetilde{K}(\mathbf{x}^i, \mathbf{x}^i)| \gg |\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j)|$. Additional knowledge on the domain of $\boldsymbol{\alpha}$ helps to refine the bound. Since $\boldsymbol{\alpha} \in \mathbb{R}^n_+$, we can see that $\rho > 0$ and $\upsilon = \max_{i,j} |K(\mathbf{x}^i, \mathbf{x}^j)| \Rightarrow \widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) \geq 0 \Rightarrow \forall\boldsymbol{\alpha} \in \mathbb{R}^n,\ \sum_{i,j} \boldsymbol{\alpha}_i\boldsymbol{\alpha}_j\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) \geq 0$. □

One may want to normalize the newly constructed distance kernel $\widetilde{K}$, or construct a Mercer distance kernel directly from a distance metric. Let $d(\mathbf{x}^i, \mathbf{x}^j)$ represent an arbitrary distance metric between the two vectors $\mathbf{x}^i$ and $\mathbf{x}^j$, and let's define:

$$\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{\rho + \upsilon_2 d(\mathbf{x}^i, \mathbf{x}^j)}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)} \tag{9}$$

Lemma 2. *$\exists\,\upsilon, \upsilon_2,\, \rho \in \mathbb{R}$ such that Eqn. (9) produces a valid Mercer kernel and $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) \in [0,\, 1]$.*
Proof sketch. We will parametrize $\rho = \upsilon_2\upsilon{-}\rho_2$. Then Eqn. (9) becomes:

$$\begin{aligned} \widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) &= \frac{\rho + \upsilon_2 d(\mathbf{x}^i, \mathbf{x}^j)}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)} = \frac{\upsilon_2\upsilon - \rho_2 + \upsilon_2 d(\mathbf{x}^i, \mathbf{x}^j)}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)} \\ &= \frac{\upsilon_2\big(\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)\big) - \rho_2}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)} = \upsilon_2 - \frac{\rho_2}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)} \end{aligned} \tag{10}$$

We recognize the two transformations defined in Eqn. (7). We find υ and ρ_2 such that $\frac{\rho_2}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)}$ is a Mercer similarity kernel, then we find υ_2 such that $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j)$ is a Mercer distance kernel and we scale ρ_2 and υ_2 such that $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) \in [0,\, 1]$. □

Note that in many cases a simpler version of Eqn. (9) can be used by setting $\rho = 0$ and $\upsilon_2 = 1$, allowing a grid search only with respect to υ with the kernel $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{d(\mathbf{x}^i, \mathbf{x}^j)}{\upsilon + d(\mathbf{x}^i, \mathbf{x}^j)}$.

IV. Experiments And Results

We evaluated our MWC formulation using the following datasets: "adult" (adu.), "webpage" (web.), "cod-rna" (cod.), and "splice" (spl.) from [31]. The adult dataset is composed of nine partitions: a1a → a9a and the webpage dataset is composed of eight partitions: w1a → w8a. For both datasets, each partition starts with a small training set and a large testing one, and ends with a large training set and a small testing one. We run our experiments on each partition. Implementation-wise, we used the standard LIBSVM code wrapped in the OpenCv library with the RBF kernel $K_{RBF}(\mathbf{x}^i, \mathbf{x}^j) = e^{-\gamma\|\mathbf{x}^i - \mathbf{x}^j\|^2}$, and we used the default parameters (C = 1). For our approach we adapted the SMO ν-SVM algorithm of [23] with the MC dynamics from [26] and we used $\widetilde{K}(\mathbf{x}^i, \mathbf{x}^j) = 1 - K_{RBF}(\mathbf{x}^i, \mathbf{x}^j)$, $\varepsilon = 10^{-5}$. It can be shown that even with our formulation we can still compute the bias as $b = \sum_{i,j} \boldsymbol{\alpha}_i^\star \boldsymbol{\alpha}_j^\star \mathbf{y}_j \widetilde{K}(\mathbf{x}^i, \mathbf{x}^j)$, and that the decision function is $D(\mathbf{x}) = b - \sum_i \boldsymbol{\alpha}_i^\star \mathbf{y}_i \widetilde{K}(\mathbf{x}^i, \mathbf{x})$. While $\nu > 100$ gives similar results as LIBSVM, we are interested in taking advantage of the sparsity constraint and exploring what is the minimum number of support vectors needed to maintain a comparable accuracy. Hence, we set $\nu = 0.1$. We initialize our system with ≤ 10 random support vectors. We also used the same grid search ($\gamma \in [0, 1]$, $\gamma_{\text{incr.}} = 10^{-5}$) for both methods for fair comparison. Table I shows the baseline on the aforementioned datasets and Table II includes the results of our algorithm. While our solution provides comparable accuracy, it is much sparser in terms of numbers of support vectors and only requires a few iterations to converge. This leads to a training time an order of magnitude smaller than the standard LIBSVM approach, as illustrated Fig. (7).

	# feat.	train. size	test. size
adu.	123	1605 → 32561	30956 → 16281
web.	300	2477 → 49749	47272 → 14951
cod.	8	59535	271617
spl.	60	1000	2175

TABLE I

DATASET SPECIFICATIONS. THE ADU. AND WEB. DATASETS ARE PARTITIONED IN INCREASING TRAINING SET SIZE AND DECREASING TESTING SET SIZE. THE SYMBOL "$a \to b$" INDICATES THE SIZE RANGE. THE SAME TERMINOLOGY WILL BE USED FOR TABLE II.

	adu.	web.	cod.	spl.
acc. LIBSVM (%)	**79.5 ± 0.5**	**97.7 ± 0.3**	66.7	83.6
acc. ours (%)	79.1 ± 1	96.9 ± 0.3	**69.8**	83.6
#SV LIBSVM	200	∼200	200	200
#SV ours	**13.5 ± 4**	**7.75 ± 4**	**18**	**136**
t.t. LIBSVM (ms)	28.8 → 683.8	92.9 → 2343.7	176.6	22.9
t.t. ours (ms)	**0.77 → 44.4**	**5.41 → 65.6**	**31.2**	**3.71**

TABLE II

ACCURACY (ACC.), NUMBER OF SUPPORT VECTORS (#SV) AND TRAINING TIME (T.T.) FOR THE BEST PARAMETER γ ON SVM BINARY CLASS DATASETS FOR LIBSVM AND OUR APPROACH.

V. CONCLUSION AND FUTURE WORK

In this work, we demonstrated that it is possible to reformulate QP problems using directly graph theory problem transformations. Our maximum clique formulation of SVM has similar accuracy as the original formulation. At the same time, it provides a significantly smaller training time and number of support vectors. The main advantage of graph theory QP reformulation is two-fold: (1) it allows the use of any MWC algorithm to train support vectors, and (2) it applies SMO-like algorithms to solve other application formulated as a MWC. This application can also be generalized to other graph problems, including formulating support vector clustering as a maximum flow problem or using weighted k-mean for spectral clustering. This also applies to a formulation to a more general problem, as shown in the previous section. In the future, we plan to tackle other applications including multi-class SVM, image and video segmentation, and document summarization. Next, we will briefly explain how to adapt our model to these new applications.

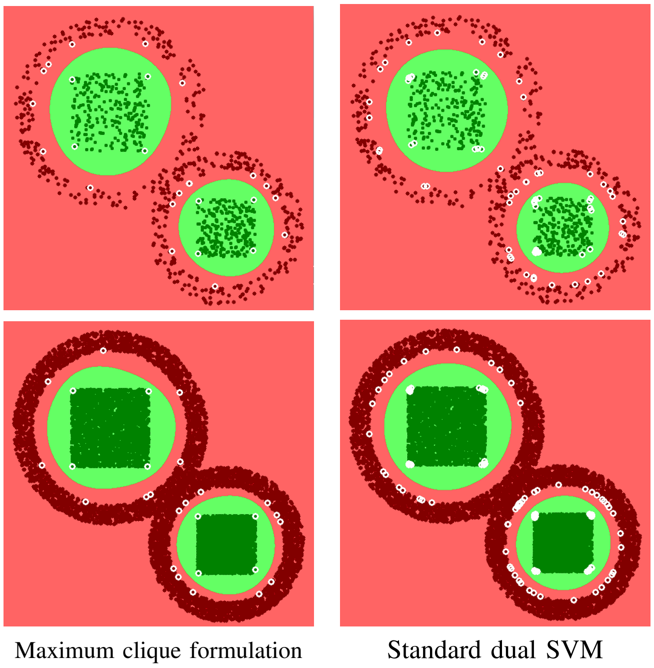


Fig. 6. 2D toy example to show the sparsity of the solution with our MWC approach (left) compared to the standard LIBSVM (right) implementation. Support vectors are circled in white. Training data is in dark green and red. Classification results are in lighter green and red. Best viewed in color.

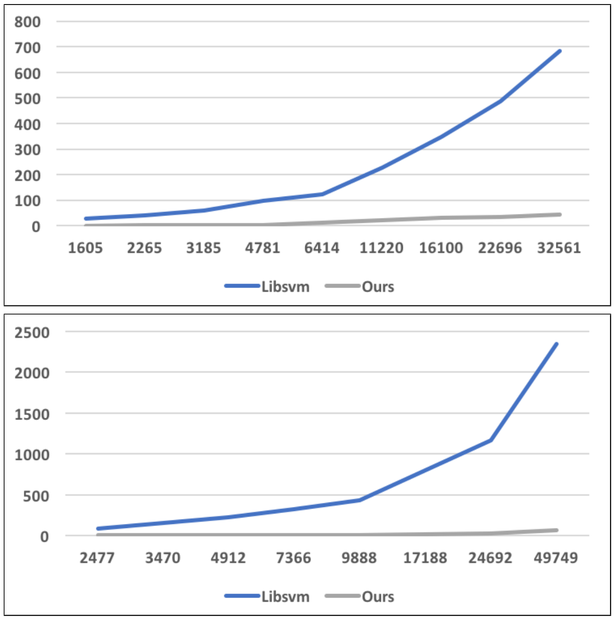


Fig. 7. Training time in ms (Y-axis) on the *adult* (top) and *webpage* (bottom) datasets: LIBSVM vs. our solution. X-axis is the dataset partition size.

So far we presented graph theory formulation of SVM for the binary classification case. In the remaining of this section we show how we can extend the MWC formulation to the multi-class problem. Following our SVM interpretation, if two support vectors are similar and are from the same class, or dissimilar and from different classes, they are positively correlated. Otherwise, they should repulse each other. We modify the edge weight by replacing the product $\mathbf{y}_i\mathbf{y}_j$ with $2\delta(\mathbf{y}_i,\, \mathbf{y}_j)$-$1$, where $\delta(\mathbf{y}_i,\, \mathbf{y}_j) = 1$ if $\mathbf{y}_i = \mathbf{y}_j$, 0 otherwise. We also extend the balancing constraint $\mathbf{y}^\intercal \boldsymbol{\alpha}$ by substituting it with $\sum_{y_i=c} \boldsymbol{\alpha}_i \leq \frac{\nu}{k}$ for every class c, where k is the total number of classes. We lose the equality as some classes need more support vectors, depending on the surrounding vectors of other classes. Indeed, pairs $\{c_l, c_m\}$ of classes still need to be exactly balanced, but not all support vectors from class c_l will contribute to the boundary between class c_l and c_m, as they may contribute to the boundary between c_l and an other one than c_m. Then final multi-class MWC formulation can be summarized as:

$$\begin{aligned} \boldsymbol{\alpha}^\star \leftarrow \arg\max_{\boldsymbol{\alpha}} \quad & \mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\intercal \widetilde{\mathbf{H}} \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{C}\boldsymbol{\alpha} \leq \frac{\nu}{k}\mathbf{1},\ \ \mathbf{C} \in I\!R^{k\times n} \ \text{ and } \ \boldsymbol{\alpha} \in [0,\, 1]^n \\ \text{with} \quad & \widetilde{\mathbf{H}}_{ij} = \Big(2\,\delta(\mathbf{y}_i,\, \mathbf{y}_j) - 1\Big)\widetilde{K}(\mathbf{x}^i,\, \mathbf{x}^j) \\ & \mathbf{y} \in \{1,\, 2, ...,\, k\}^n \ \text{ and } \ \mathbf{x}^i \in I\!R^d \end{aligned} \tag{11}$$

where $\mathbf{C}_{ij} = \delta(\mathbf{y}_j,\, i)$, i.e. $\mathbf{C}_{ij} = 1$ if $\mathbf{y}_j = i$, 0 otherwise. Note that the extension could have also been applied in the original formulation Eqn. (1), and that the same ν-SVM algorithm of [23] can easily be adapted to the multi-class problem. The initialization strategy described in Section III-B holds as well for the multi-class problem.

VI. Acknowledgements

This work would not have been possible without the guidance and support of David Salesin, Joel Brandt, and Bobby Murphy. We would also like to thank Melanie Poblete and Aletta Hiemstra for their help in proofreading the document and the anonymous reviewers for their useful comments.

References

[1] N. Deo, *Graph Theory with Applications to Engineering and Computer Science (Prentice Hall Series in Automatic Computation)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1974.

[2] R. P. Singh and Vandana, "Article: Application of graph theory in computer science and engineering," *International Journal of Computer Applications*, vol. 104, no. 1, pp. 10–13, October 2014.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, 2000.

[4] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 29, p. 2007, 2007.

[5] W. Brendel and S. Todorovic, "Segmentation as maximum-weight independent set." in *NIPS*, 2010.

[6] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE TPAMI*, 2007.

[7] E. Zemene and M. Pelillo, "Interactive image segmentation using constrained dominant sets," *CoRR*, vol. abs/1608.00641, 2016.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, 2004.

[9] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recogn.*, vol. 46, no. 3, pp. 1020–1038, Mar. 2013.

[10] W. Brendel, M. R. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *CVPR*, 2011, pp. 1273–1280.

[11] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," ser. ECCV'12, 2012, pp. 343–356.

[12] A. Dehghan, S. Modiri Assari, and M. Shah, "Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *CVPR*, June 2015.

[13] L. Liu, T. G. Dietterich, N. Li, and Z. Zhou, "Transductive optimization of top k precision," *CoRR*, vol. abs/1510.05976, 2015.

[14] S. G. Vadlamudi, S. Sengupta, S. Kambhampati, M. Taguinod, Z. Zhao, A. Doupé, and G. Ahn, "Moving target defense for web applications using bayesian stackelberg games," *CoRR*, vol. abs/1602.07024, 2016.

[15] X. Wu, D. Sheldon, and S. Zilberstein, "Optimizing resilience in large scale networks," in *Proceedings of the Thirtieth Conference on Artificial Intelligence*, 2016.

[16] Q. Zhou, W. Chen, S. Song, J. R. Gardner, K. Q. Weinberger, and Y. Chen, "A reduction of the elastic net to support vector machines with an application to gpu computing," in *Proceedings of AAAI*, 2015, pp. 3210–3216.

[17] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15, 2015, pp. 2153–2159.

[18] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *CVPR'15*.

[19] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *J. Mach. Learn. Res.*, vol. 6, pp. 363–392, Dec. 2005.

[20] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *JMLR'01*.

[21] W. W. Hager and J. T. Hungerford, "Continuous quadratic programming formulations of optimization problems on graphs," *European Journal of Operational Research*, vol. 240, no. 2, pp. 328–337, 2015.

[22] W. W. Hager, D. T. Phan, and H. Zhang, "An exact algorithm for graph partitioning," *CoRR'09*.

[23] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, Dec. 2005.

[24] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Advances in Kernel Methods - Support Vector Learning, Tech. Rep., 1998.

[25] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, Jan. 2005.

[26] S. R. Bul and I. M. Bomze, "Infection and immunization: A new class of evolutionary game dynamics." *Games and Economic Behavior*, vol. 71, no. 1, pp. 193–211, 2011.

[27] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[28] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *COLT workshop*, 1992.

[29] T. S. Motzkin and E. G. Straus, "Maxima for graphs and a new proof of a theorem of Turán," *Canadian Journal of Mathematics*, vol. 17, pp. 533–540, 1965.

[30] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[31] C.-C. Chang and C.-J. Lin, "www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html," LIBSVM Data: Binary Classification.