

A Graph Theory Approach To QP Problem Reformulation: An Example With SVM

William Brendel

Snap Inc. Research

Luis Marujo

Snap Inc. Research

william.brendel@snap.com

luis.marujo@snap.com

Abstract

Quadratic programming (QP) problem reformulations have been studied for decades [Sherali and Tuncbilek(1995), Nemirovski and Shapiro(2006), Anstreicher(2009)], [Zheng et al.(2012)Zheng, Sun, Li, and Cui, Wu and Jiang(2017)], but rarely linked to Graph Theory. Indeed, typical reformulations focus on convexifying a non-convex QP problem, making the objective function differentiable, optimizing on the continuous domain while ensuring the final solution is binary, or adding regularizers and Lagrangian coefficients to optimize the dual problem. In this paper, we take SVM as an example to 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 types of reformulation. We show that SVM is comparable to a soft weighted edge independent set problem where the amount of support vectors per class is balanced, thus it can also be reformulated as a maximum weighted clique problem (MWC) with the same class balancing constraint. After adapting the sequential minimal optimization (SMO) algorithm [Platt(1998), Fan et al.(2005)Fan, Chen, and Lin] to our new MWC formulation, we demonstrate that such reformulation leads to improved performance (7~36 time faster to train, and sparser solution for comparable accuracy).

1 Introduction

In the fields of Machine Learning, Natural Language Processing, Computer Graphics, and Computer Vision, problems involving quadratic programming (QP) formulation are abundant. It is not surprising, since many applications 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. For the last 50 years, graph theory has simply been supporting an ocean of computer science applications [Deo(1974), Singh and Vandana(2014)]. To name a few examples, image segmentation as been formulated as minimum cut [Shi and Malik(2000), Dhillon et al.(2007)Dhillon, Guan, and Kulis], maximum weight independent set [Brendel and Todorovic(2010)], maximum weight clique [Pavan and Pelillo(2007), Zemene and Pelillo(2016)] and minimum spanning tree [Felzenszwalb and Huttenlocher(2004)] problems [Peng et al.(2013)Peng, Zhang, and Zhang]. Multi-object tracking has been formulated as maximum weight independent set [Brendel et al.(2011)Brendel, Amer, and Todorovic] and generalized minimum and maximum clique [Zamir et al.(2012)Zamir, Dehghan, and Shah, Dehghan et al.(2015)Dehghan, Modiri Assari, and Shah] problems. Since graph problems can be reduced to one another, QP problems can also be reformulated into problems that can be solved more efficiently. Various cut problems are reformulated as spectral clustering problems that can be optimized via weighted kernel k-means algorithms [Dhillon et al.(2007)Dhillon, Guan, and Kulis, Li and Chen()], achieving real-time computation performances. In Tsang et al.'s work [Tsang et al.(2005)Tsang, Kwok, and Cheung], SVM and Support Vector Clustering (SVC) are formulated as minimum enclosing ball problems, obtaining provably approximate optimal solutions in linear time, with a space complexity independent of the problem size. This work demonstrates how SVM formulation [Ben-Hur et al.(2000)Ben-Hur, Horn, Siegelmann, and Vapnik] can be viewed as an independent set problem [Tsang et al.(2005)Tsang, Kwok, and Cheung, Hager and Hungerford(2015)], and thus be reduced to other graph problems.

Review of SVM Dual Formulation And SMO: SVM aims at learning boundaries between feature vectors $\{\mathbf{x}^k\}_{k=1}^n$ associated to different classes $\{\mathbf{y}_k\}_{k=1}^n$. When classes are binary, the SVM dual formulation has the

following form:

$$\begin{aligned}
\boldsymbol{\alpha}^* \leftarrow \arg \max_{\boldsymbol{\alpha} \in [0, 1]^n} \mathcal{F}(\boldsymbol{\alpha}) : \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{H} \boldsymbol{\alpha} \\
\text{s.t. } \mathbf{y}^\top \boldsymbol{\alpha} = 0 \quad \text{with } \mathbf{y} \in \{-1, 1\}^n \\
\mathbf{H}_{ij} = \mathbf{y}_i \mathbf{y}_j K(\mathbf{x}^i, \mathbf{x}^j) \quad \text{and } \mathbf{x}^k \in \mathbb{R}^d
\end{aligned} \tag{1}$$

where $\alpha_k > 0$ 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 $\{\mathbf{x}^k \mid \alpha_k > 0\}$ in a vector space where the dot product is induced by the kernel function K . SMO was originally designed to train a support vector machine that requires the solution of the very large SVM dual QP optimization problem [Platt(1998)] defined above. SMO belongs to the family of SQP algorithms [Gill et al.(2005)Gill, Murray, and Saunders]. 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.

Contributions: In this work we show how to reformulate a QP problem using graph theory. More precisely, we relate 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, in 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 [Pavan and Pelillo(2007), Zemene and Pelillo(2016)], the later being used to define the maximum weighted clique (MWC) problem [Hager et al.(2015)Hager, Phan, and Zhang, Hager and Hungerford(2015)]. We show that our MWC formulation involves Mercer distance kernels, instead of Mercer similarity kernels, and demonstrate how to construct such kernel, leading to new families of kernels. Finally, we show that our MWC formulation has computational advantages while preserving comparable accuracy compared to the standard LIBSVM implementation of the SVM dual QP formulation.

2 Problem Setup And Reformulation

The independent set QP formulation has the form of $\mathbf{x}^* \leftarrow \arg \max_{\mathbf{x} \in \{0, 1\}^n} \mathbf{1}^\top \mathbf{x} - \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x}$, where \mathbf{A} is a matrix representing edge connectivity in a graph $\mathcal{G}(V, E)$, i.e., edge $(i, j) \in E \Leftrightarrow \mathbf{A}_{ij} > 0$ [Hager and Hungerford(2015)]. Typical algorithms loosen the binary constraint $\mathbf{x} \in \{0, 1\}^n$ to the continuous domain $\mathbf{x} \in [0, 1]^n$, and add a regularization diagonal matrix \mathbf{D} : $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{D}$, such that the final solution entries \mathbf{x}_i^* are push to the domain boundaries [Hager et al.(2015)Hager, Phan, and Zhang]. We can clearly see that Eqn. (1) resembles to the independent set QP formulation, with two differences. First $\mathbf{H} \in \mathbb{R}^{n \times n}$ whereas $\mathbf{A} \in \{0, 1\}^n$, and second we have an additional constraint $\mathbf{y}^\top \boldsymbol{\alpha} = 0$. One can show that: on one side remapping $\mathbf{A} \leftarrow 2\mathbf{A} - \mathbf{1}\mathbf{1}^\top$ so that $\mathbf{A} \in \{-1, 1\}$ does not affect the solution of the independent set QP as it simply adds an additional maximizer $(\mathbf{1}^\top \mathbf{x})^2$ to the objective function; on the other side we could normalize the Kernel to produce values $K(\mathbf{x}^i, \mathbf{x}^j) \in [0, 1]$ so that $\mathbf{H} \in [-1, 1]^{n \times n}$. Support vectors are then viewed as nodes in \mathcal{G} connected to each other with a soft correlation weight on edges: 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. The additional constraint $\mathbf{y}^\top \boldsymbol{\alpha} = 0$ imposes the solution to be balanced: the amount of support vectors for each class should be the same. Now if SVM is viewed as an independent set problem, it can also be reduced to an MWC problem.

2.1 SVM As A Maximum Weighted Clique Problem

The dual SVM formulation is the same as the maximum weighted independent (MWI) set formulation [Hager and Hungerford(2015)], with the additional constraint $\mathbf{y}^\top \boldsymbol{\alpha} = 0$ and where the graph edge weights have soft values (i.e. \mathbf{H}_{ij} in Eqn. (1)). We can switch from the MWI problem to the MWC problem by taking the complement of the graph used for the independent set problem. In other words we need to solve:

$$\begin{aligned}
\boldsymbol{\alpha}^* \leftarrow \arg \max_{\boldsymbol{\alpha}} \mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^\top \tilde{\mathbf{H}} \boldsymbol{\alpha} \\
\text{s.t. } \mathbf{y}^\top \boldsymbol{\alpha} = 0 \quad \text{and } \mathbf{1}^\top \boldsymbol{\alpha} = \nu \\
\boldsymbol{\alpha} \in [0, 1]^n \quad \text{and } \mathbf{y} \in \{-1, 1\}^n \\
\text{with } \tilde{\mathbf{H}}_{ij} = \mathbf{y}_i \mathbf{y}_j \tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \quad \text{and } \mathbf{x}^k \in \mathbb{R}^d
\end{aligned} \tag{2}$$

where $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j)$ is now a Mercer kernel representing the distance between the two vectors \mathbf{x}^i and \mathbf{x}^j instead of their similarity. Here ν controls the sparsity of the solution. Next, we demonstrate how to construct a proper Mercer distance kernel.

2.2 Mercer Distance Kernels

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

$$\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{\rho}{v + K(\mathbf{x}^i, \mathbf{x}^j)} \quad \text{or} \quad \tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = v - K(\mathbf{x}^i, \mathbf{x}^j) \quad (3)$$

Note that other transformation functions like $x \rightarrow e^{-\nu x^\rho}$ can be used as well. Proof and new kernel construction can be found in [Shawe-Taylor and Cristianini(2004)]. Interestingly enough, we can build a similarity measure from a distance metric the same way.

Lemma 1. $\exists \rho, v \in \mathbb{R}$ such that Eqn. (3) always produces a valid Mercer kernel.

Proof sketch. We base the proof on the Gershgorin circle theorem. We can find v and ρ such that:

$$\min \text{ eigenvalue} \geq \min_i \left(\tilde{K}(\mathbf{x}^i, \mathbf{x}^i) - \sum_{j \neq i} \tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \right) \geq 0 \quad (4)$$

However the Gershgorin bound is very loose, and solving Eqn. (4) will produce a final matrix where $|\tilde{K}(\mathbf{x}^i, \mathbf{x}^i)| \gg |\tilde{K}(\mathbf{x}^i, \mathbf{x}^j)|$. Additional knowledge on the domain of α helps to refine the bound. Since $\alpha \in \mathbb{R}_+^n$ we can see that $\rho > 0$ and $v = \max_{i,j} |K(\mathbf{x}^i, \mathbf{x}^j)| \Rightarrow \tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \geq 0 \Rightarrow \forall \alpha \in \mathbb{R}^n, \sum_{i,j} \alpha_i \alpha_j \tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \geq 0$. \square

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

$$\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{\rho + v_2 d(\mathbf{x}^i, \mathbf{x}^j)}{v + d(\mathbf{x}^i, \mathbf{x}^j)} \quad (5)$$

Lemma 2. $\exists v, v_2, \rho \in \mathbb{R}$ such that Eqn. (5) produces a valid Mercer kernel and $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \in [0, 1]$.

Proof sketch. Let's parametrize $\rho = v_2 v - \rho_2$. Then Eqn. (5) becomes:

$$\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{\rho + v_2 d(\mathbf{x}^i, \mathbf{x}^j)}{v + d(\mathbf{x}^i, \mathbf{x}^j)} = \frac{v_2 v - \rho_2 + v_2 d(\mathbf{x}^i, \mathbf{x}^j)}{v + d(\mathbf{x}^i, \mathbf{x}^j)} = v_2 - \frac{\rho_2}{v + d(\mathbf{x}^i, \mathbf{x}^j)} \quad (6)$$

We recognize the two transformations defined in Eqn. (3). We find v and ρ_2 such that $\frac{\rho_2}{v + d(\mathbf{x}^i, \mathbf{x}^j)}$ is a Mercer similarity kernel, then we find v_2 such that $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j)$ is a Mercer distance kernel and we scale ρ_2 and v_2 such that $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) \in [0, 1]$. \square

Note that in many cases a simpler version of Eqn. (5) can be used by setting $\rho = 0$ and $v_2 = 1$, allowing a grid search only with respect to v with the kernel $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = \frac{d(\mathbf{x}^i, \mathbf{x}^j)}{v + d(\mathbf{x}^i, \mathbf{x}^j)}$.

3 Experiments And Results

We evaluated our MWC formulation using the following datasets: “adult” (adu.), “webpage” (web.), “cod-rna” (cod.), and “splice” (spl.) from [Chang and Lin(LIBSVM Data: Binary Classification)]. The adult dataset is composed of nine partitions: a1a \rightarrow a9a and the webpage dataset is composed of eight partitions: w1a \rightarrow 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 $K_{RBF}(\mathbf{x}^i, \mathbf{x}^j) = e^{-\gamma \|\mathbf{x}^i - \mathbf{x}^j\|^2}$. For our approach we adapted the SMO ν -SVM algorithm of [Fan et al.(2005)Fan, Chen, and Lin] and we used $\tilde{K}(\mathbf{x}^i, \mathbf{x}^j) = 1 - K_{RBF}(\mathbf{x}^i, \mathbf{x}^j)$, $\varepsilon = 10^{-5}$. It can be shown that even with our formulation we can still compute the bias as $b = \sum_{i,j} \alpha_i^* \alpha_j^* \mathbf{y}_j \tilde{K}(\mathbf{x}^i, \mathbf{x}^j)$, and that the decision function is $D(\mathbf{x}) = b - \sum_i \alpha_i^* \mathbf{y}_i \tilde{K}(\mathbf{x}^i, \mathbf{x})$. 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

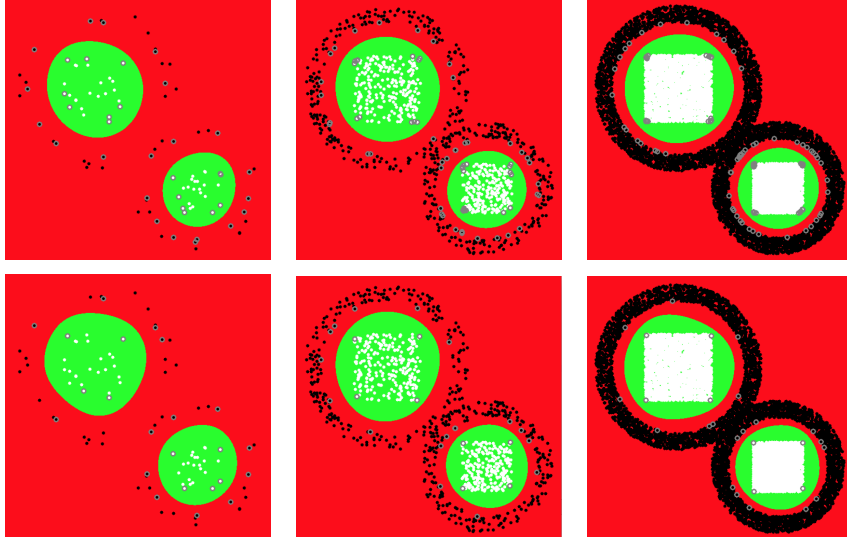


Figure 1: Toy example (100, 1000, 10000 data points) to show the sparsity of the solution with our MWC approach (bottom row) compared to the standard LIBSVM (top row) implementation. Support vectors are circled in gray. Training data is in white and black. Classification results are in green and red. Best viewed in color.

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 1 shows the baseline on the aforementioned datasets and Table 2 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.

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

Table 1: Dataset specifications. the adu. and web. datasets are partitioned in increasing training set size and decreasing testing set size. The symbol “ $a \rightarrow b$ ” indicates the size range. The same terminology will be used for Table 2.

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

Table 2: 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.

4 Conclusion

In this work, we showed that we can reformulate QP problems using graph theory problem transformations directly. Our maximum clique formulation of SVM produces similar accuracy than the original formulation, and provides significantly smaller training time and smaller number of support vectors. The main advantage of graph theory QP reformulation is a two way street: while now we can use any MWC algorithm to train support vectors, we can use SMO-like algorithm to solve other application formulated as a MWC. This generalizes to other graph problems as well, like formulating support vector clustering as a maximum flow problem, or using weighted k-mean for spectral clustering. This also allows to extend a formulation to a more general problem: for example by replacing the product $\mathbf{y}_i \mathbf{y}_j$ by $2\delta(\mathbf{y}_i, \mathbf{y}_j) - 1$ and the constraint $\mathbf{y}^T \boldsymbol{\alpha}$ by $\sum_{y_i=c} \alpha_i = k$ in Eqn. (1) or Eqn. (2), we have now a multi-class SVM formulation. In the future, we plan to tackle other applications, like multi-class SVM, image and video segmentation and, document summarization.

References

- [Anstreicher(2009)] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *J. of Global Optimization*, 2009. ISSN 0925-5001. doi: 10.1007/s10898-008-9372-0. URL <http://dx.doi.org/10.1007/s10898-008-9372-0>.
- [Ben-Hur et al.(2001)] Ben-Hur, Horn, Siegelmann, and Vapnik] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *JMLR'01*.
- [Brendel and Todorovic(2010)] W. Brendel and S. Todorovic. Segmentation as maximum-weight independent set. In *NIPS*, 2010. URL <http://dblp.uni-trier.de/db/conf/nips/nips2010.html#BrendelT10>.
- [Brendel et al.(2011)] Brendel, Amer, and Todorovic] W. Brendel, M. R. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, pages 1273–1280, 2011. doi: 10.1109/CVPR.2011.5995395. URL <http://dx.doi.org/10.1109/CVPR.2011.5995395>.
- [Chang and Lin(LIBSVM Data: Binary Classification)] C.-C. Chang and C.-J. Lin. www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html, LIBSVM Data: Binary Classification.
- [Dehghan et al.(2015)] Dehghan, Modiri Assari, and Shah] 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.
- [Deo(1974)] N. Deo. *Graph Theory with Applications to Engineering and Computer Science (Prentice Hall Series in Automatic Computation)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1974. ISBN 0133634736.
- [Dhillon et al.(2007)] Dhillon, Guan, and Kulis] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:2007, 2007.
- [Fan et al.(2005)] Fan, Chen, and Lin] 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.*, 6:1889–1918, Dec. 2005. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1046920.1194907>.
- [Felzenszwalb and Huttenlocher(2004)] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.
- [Gill et al.(2005)] Gill, Murray, and Saunders] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Rev.*, 47(1):99–131, Jan. 2005. ISSN 0036-1445. doi: 10.1137/S0036144504446096. URL <http://dx.doi.org/10.1137/S0036144504446096>.
- [Hager and Hungerford(2015)] W. W. Hager and J. T. Hungerford. Continuous quadratic programming formulations of optimization problems on graphs. *European Journal of Operational Research*, 240(2):328–337, 2015. URL <http://EconPapers.repec.org/RePEc:eee:ejores:v:240:y:2015:i:2:p:328-337>.
- [Hager et al.(2009)] Hager, Phan, and Zhang] W. W. Hager, D. T. Phan, and H. Zhang. An exact algorithm for graph partitioning. *CoRR'09*.
- [Li and Chen()] Z. Li and J. Chen. Superpixel segmentation using linear spectral clustering. In *CVPR2015*.
- [Nemirovski and Shapiro(2006)] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM J. on Optimization*, 17(4):969–996, Dec. 2006. ISSN 1052-6234. doi: 10.1137/050622328. URL <http://dx.doi.org/10.1137/050622328>.
- [Pavan and Pelillo(2007)] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE TPAMI*, 2007.
- [Peng et al.(2013)] Peng, Zhang, and Zhang] B. Peng, L. Zhang, and D. Zhang. A survey of graph theoretical approaches to image segmentation. *Pattern Recogn.*, 46(3):1020–1038, Mar. 2013. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.09.015. URL <http://dx.doi.org/10.1016/j.patcog.2012.09.015>.
- [Platt(1998)] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in Kernel Methods - Support Vector Learning, 1998.
- [Shawe-Taylor and Cristianini(2004)] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972.
- [Sherali and Tuncbilek(1995)] H. D. Sherali and C. H. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7(1):1–31, Jul 1995.
- [Shi and Malik(2000)] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8), 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL <http://dx.doi.org/10.1109/34.868688>.
- [Singh and Vandana(2014)] R. P. Singh and Vandana. Article: Application of graph theory in computer science and engineering. *International Journal of Computer Applications*, 104(1):10–13, October 2014.
- [Tsang et al.(2005)] Tsang, Kwok, and Cheung] 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.*, 6:363–392, Dec. 2005. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1046920.1058114>.
- [Wu and Jiang(2017)] B. Wu and R. Jiang. Quadratic convex reformulations for integer and mixed-integer quadratic programs. pages 43–58, 01 2017.
- [Zamir et al.(2012)] Zamir, Dehghan, and Shah] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. *ECCV'12*, pages 343–356, 2012. doi: 10.1007/978-3-642-33709-3_25. URL http://dx.doi.org/10.1007/978-3-642-33709-3_25.
- [Zemene and Pelillo(2016)] E. Zemene and M. Pelillo. Interactive image segmentation using constrained dominant sets. *CoRR*, abs/1608.00641, 2016. URL <http://arxiv.org/abs/1608.00641>.
- [Zheng et al.(2012)] Zheng, Sun, Li, and Cui] X. Zheng, X. Sun, D. Li, and X. Cui. Lagrangian decomposition and mixed-integer quadratic programming reformulations for probabilistically constrained quadratic programs. *European Journal of Operational Research*, 221(1):38 – 48, 2012. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2012.03.006>. URL <http://www.sciencedirect.com/science/article/pii/S0377221712001993>.