Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Lifted Inference for Convex Quadratic Programs

Martin Mladenov
TU Dortmund University, Germany
martin.mladenov@cs.tu-dortmund.de

Leonard Kleinhans
TU Dortmund University, Germany
leonard.kleinhans@cs.tu-dortmund.de

Kristian Kersting
TU Dortmund University, Germany
kristian.kersting@cs.tu-dortmund.de

Abstract

Symmetry is the essential element of lifted inference that has recently demonstrated the possibility to perform very efficient inference in highly-connected, but symmetric probabilistic models. This raises the question, whether this holds for optimization problems in general. Here we show that for a large class of optimization methods this is actually the case. Specifically, we introduce the concept of fractional symmetries of convex quadratic programs (QPs), which lie at the heart of many AI and machine learning approaches, and exploit it to lift, i.e., to compress QPs. These lifted QPs can then be tackled with the usual optimization toolbox (off-the-shelf solvers, cutting plane algorithms, stochastic gradients etc.). If the original QP exhibits symmetry, then the lifted one will generally be more compact, and hence more efficient to solve.

Introduction

Convex optimization is arguably one of the main motors behind Artificial Intelligence (AI) as it enables inference and learning in a wide variety of AI models, such as SVMs, LASSO and efficient approximations (e.g. variational approaches, convex NMF) to hard inference tasks. The language in which convex optimization problems are specified includes inequalities, matrix and tensor algebra, and software packages for convex optimization such as CVXPY (Diamond, Chu, and Boyd 2014) recreate this language as an interface between the user and the solver. Unfortunately, a pure algebraic language has one shortcoming: it is difficult—if not impossible—for the non-expert to directly make use of the discrete, combinatorial structure often underlying convex programs; pixels depend only on neighboring pixels; the reward of placing a cup on a table does not depend on whether the window in the next room is open. Having a richer representation such as first-order logic to express the combinatorial structure and an automatic way to utilize it in the solver, however, is likely extend the reach and efficiency of AI. This has been demonstrated by statistical relational learning (SRL) that has argued in favor of first-order languages when dealing with complex graphical models, see e.g. (De Raedt et al. 2016) for a recent overview. Due to the high-level nature of the relational probabilistic languages, the low-level (ground) model they produce might often contain redundancies in terms of symmetries. Lifted probabilistic inference approaches (Poole 2003) exploit these symmetries to perform very efficient inference in highly-connected (and hence otherwise often intractable for traditional inference approach) but symmetric models. Intuitively, one infers which variables are indistinguishable in the ground model (if possible without actually grounding) and solves the model treating the indistinguishable variables as groups instead of individuals to reduce the dimensionality of the model. Unfortunately, SRL does not support convex quadratic programs.

Here, we demonstrate that the core idea of SRL can be transferred to convex quadratic programming. As our main contribution, we formalize the notion of symmetries of convex quadratic programs (QPs). Specifically, we first show that unlike for graphical models, where the notion of indistinguishability of variables is that of exact symmetry (automorphisms of the factor graph), QPs admit a weaker (partitions of indistinguishable variables which are at least as coarse) notion of indistinguishability called a fractional automorphism (FA) resp. equitable partition (EP) computable in quasi-linear time. This implies that more general lifted inference rules for QPs can be designed. This is surprising, as it was believed that FAs apply only to linear equations. Second, we investigate geometrically how FAs of quadratic forms arise. The existing theory of symmetry in convex quadratic forms states that an automorphism of $\boldsymbol{x}^T Q \boldsymbol{x}$ corresponds to a rotational symmetry of the semidefinite factors of Q. We generalize this in that FA of $\boldsymbol{x}^T Q \boldsymbol{x}$ can be related not only to rotations, but also to certain scalings. This then results in the first approximate FA approach based on standard clustering techniques and whitening. Finally, we tackle the question to which extend kernels might preserve fractional symmetry. All this is embedded in the first relational QP language as illustrated in Fig. 1(left), which is not discussed due to space limitations.

In doing so, the present work is the first that introduces relational convex QPs and studies their symmetries. Indeed, there are symmetry-breaking branch-and-bound approaches for (mixed–)integer programming (Margot 2010) that are also featured by commercial solvers. QPs, however, do not feature branch-and-bound solvers. For the special fragment of LPs, (Kersting, Mladenov, and Tokmakov 2015) have introduced a relational language and shown how to exploit fractional symmetries. (Relaxed) graph automorphisms and variants have

```
# logical query for linking papers
linked(I1, I2) = label(I1) & query(I2) & (cite(I1, I2) | cite(I2, I1))
# inline definitions
slacks = sum{I in labeled(I)} slack(I); coslacks = sum{I1, I2 in linked(I1, I2)} slack(I1,I2)
# QUADRATIC OBJECTIVE, the main novelty compared to [Kersting et al., 2015]
minimize: sum{J in feature(I,J)} weight(J)**2 + c1 * slack + c2 * coslack;
subject to forall {I in labeled(I)}: labeled(I)*predict(I) >= 1 - slack(I); # correct prediction
subject to forall {I in labeled(I)}: slack(I) >= 0; # slacks are positive
# TRANSDUCTIVE PART: cited instances should have the same labels.
subject to forall {I1, I2 in linked(I1, I2)}: labeled(I1) * predict(I2) >= 1 - slack(I1, I2);
subject to forall {I1, I2 in linked(I1, I2)}: coslack(I1, I2) >= 0; #coslacks are positive
```

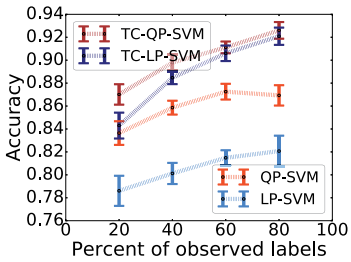

Figure 1: Illustration of the relational QP language, a novel extension of the relational LP language due to (Kersting, Mladenov, and Tokmakov 2015). (Left) A relational SVM (TC-QP-SVM) for collective classification of objects with relations among them, here scientific papers with citations. No "relational" kernel is used, just a linear one with relational constraints taking the citation links into account. (Right) QPs outperform LPs, in particular for less many observed labels. (Best viewed in color)

been explored for graph kernels (Shervashidze and Borgwardt 2009) and (I)LP-MAP inference approaches (Bui, Huynh, and Riedel 2013; Mladenov, Globerson, and Kersting 2014; Jernite, Rush, and Sontag 2015). Unfortunately, their proofs do not carry over to (convex) QPs. (Güler and Gürtuna 2012) and references in there have studied automorphisms but not fractional ones of convex sets. Finally, indeed, several expressive modeling languages for mathematical programming have been proposed, see e.g. (Wallace and Ziemba 2005) for a recent overview. They are mixtures of declarative and imperative programming styles using sets of objects to index multidimensional parameters and variables. Recently, (Diamond, Chu, and Boyd 2014) enabled an object-oriented approach to constructing optimization problems. None of them provide integrated capabilities with relational logic, not to speak of lifting. The need for relational mathematical programming languages is witnessed e.g. in natural language processing (Yih and Roth 2007; Riedel, Smith, and McCallum 2012) and the recent push to marry statistical analytic frameworks like R and Python with relational databases (Ré et al. 2015).

We proceed as follows. We start off by developing automorphisms of QPs, introducing the required background on the fly. Then, we generalize this to fractional symmetries. Before concluding, we illustrate our theoretical results empirically.

Exact Symmetries of Convex QPs

Let us start off with exact symmetries of convex QPs. **Lifting convex quadratic programs** essentially amount to reducing the size a model by grouping together "indistinguishable" variables and constraints. In other words, they exploit symmetries. To formalize the notion of lifting more concisely let us consider a **convex program**, i.e., an optimization problem of the form

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{D}} J(\boldsymbol{x}) \;, \qquad (\clubsuit)$$

over $\boldsymbol{x} \in \mathbb{R}^n$, where $J : \mathbb{R}^n \to \mathbb{R}$ is a convex function, and $\mathcal{D}$ is a subset of $\mathbb{R}^n$, typically specified as the solution a system of convex inequalities $f_1(\boldsymbol{x}) \leqslant 0, \ldots, f_m(\boldsymbol{x}) \leqslant 0$. A **convex quadratic program** (QP) is an instance of $(\clubsuit)$ where $J(\boldsymbol{x}) = \boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$ is a quadratic function with $Q \in \mathcal{R}^{n\times n}$ is symmetric and positive semi-definite, and $\mathcal{D}$ is a convex subset of $\mathcal{R}^n$ specified as a system of linear equations, $\mathcal{D} = \{\boldsymbol{x} : A\boldsymbol{x} \leqslant \boldsymbol{b}\}$. If Q is the zero matrix, the problem is known as a **linear program** (LP). If we add convex quadratic constraints to a quadratic program, we obtain a quadratically constrained quadratic program (QCQP). We will not deal explicitly with QCQPs in this paper, however, by the end of our discussion of quadratic functions, it will be evident that our results can easily be extended to such programs. We shall denote a QP by the tuple $\boldsymbol{QP} = (Q, \boldsymbol{c}, A, \boldsymbol{b})$.

We are now interested in partitioning the variables of the program by a partition $\mathcal{P} = \{P_1, \ldots, P_p\}$, $P_i \cap P_j = \varnothing$, $\bigcup_i P_i = \{x_1, \ldots, x_n\}$, such that there exists at least one solution that **respects** the partition. More formally, $\mathcal{P}$ is a **lifting partition** of $(\clubsuit)$ if $(\clubsuit)$ admits an optimal solution with $x_i = x_j$ whenever x_i and x_j are in the same class in $\mathcal{P}$. We call the linear subspace defined by the latter condition $\mathbb{R}_{\mathcal{P}}$. Having apriori obtained a lifting partition of the QP, we can restrict the solution space to $\mathcal{D}\cap\mathbb{R}_{\mathcal{P}}$. That is, we constrain indistinguishable variables to be equal, knowing that at least one solution will be preserved in this space of lower dimension. Since ground variables of the same class are now equal, they can be replaced with a single aggregated (lifted) variable. The resulting lifted problem has one variable per equivalence class, thus, if the lifting partition is coarse enough, significant dimensionality reduction and run-time savings can be achieved. To recover a ground solution, one assigns the value of the lifted variable to every ground variable in its class.

One way to demonstrate that a given partition $\mathcal{P}$ is a lifting partition for $(\clubsuit)$ is by showing that **averaging** any feasible $\boldsymbol{x}$ over the partition classes (i.e. $\widetilde{x}_i = \frac{1}{|\operatorname{class}(x_i)|}\sum_{x_j\in\operatorname{class}(x_i)} x_j$) yields a new feasible $\widetilde{\boldsymbol{x}}$ with $J(\widetilde{\boldsymbol{x}}) \leqslant J(\boldsymbol{x})$. Theorem 1 that we will prove later on will provide sufficient conditions for this. As a consequence, by averaging any optimal solution we get another optimal solution which respects $\mathcal{P}$, implying that $\mathcal{P}$ is a lifting partition. Please note if there is only one solution with different values in all coordinates then this certifies that there are no symmetries. In any case, one bit of notation that is handy in the analysis averaging operations is the **partition matrix**. To any partition $\mathcal{P}$ we can associate a matrix $X^{\mathcal{P}} \in \mathbb{Q}^{n\times n}$ such that $X^{\mathcal{P}}_{ij} = 1/|\operatorname{class}(x_i)|$ if $x_j \in \operatorname{class}(x_i)$ or 0 otherwise. With $X^{\mathcal{P}}$ defined thusly, averaging $\boldsymbol{x}$ over the classes of $\mathcal{P}$ is equivalent to multiplying by $X^{\mathcal{P}}$, i.e., $\widetilde{\boldsymbol{x}} = X^{\mathcal{P}}\boldsymbol{x}$. Partition matrices are always **doubly stochastic** ($X^{\mathcal{P}}\mathbf{1} = \mathbf{1}$), **symmetric** ($(X^{\mathcal{P}})^T = X^{\mathcal{P}}$), and **idempotent** ($X^{\mathcal{P}}X^{\mathcal{P}} = X^{\mathcal{P}}$) –

as a consequence also **semidefinite**.

Example. We seek to minimize the function $\boldsymbol{x}^T Q \boldsymbol{x}$ over $\boldsymbol{x} \in \mathbb{R}^4$, subject to $\boldsymbol{x} \geqslant 1$, with Q given in Fig. 2. As a lifting partition, we propose $\mathcal{P} = \{\{x_1, x_3\}, \{x_2, x_4\}\}$ (in the next paragraph, we will explain how one could compute this lifting partition). The corresponding partition matrix $X^{\mathcal{P}}$ is also shown on Fig. 2. Let us demonstrate that averaging over the classes of $\mathcal{P}$ decreases the value of the solution. For example, for $\boldsymbol{x}_0 = [2, 1, 1, 2]^T$, $\boldsymbol{x}_0^T Q \boldsymbol{x}_0 = 3$. On the other hand, the class-averaged $\widetilde{\boldsymbol{x}}_0 = X_{\mathcal{P}} \boldsymbol{x}_0 = [1.5, 1.5, 1.5, 1.5]^T$ yields a value of 0. In fact, one could notice that any feasible $\boldsymbol{x}$ respecting the partition yields a value of 0, so any such solution is optimal. Moreover, if all coordinates of $\boldsymbol{x}$ are already greater than or equal to 1, then the same holds for $\widetilde{\boldsymbol{x}}$, as averages cannot be lower than the minimum of the averaged numbers. Thus, the compressed problem reduces to finding any two numbers that $\geqslant 1$. □

An intuitive way to find lifting partitions is via **automorphism groups** of convex problems. We define the automorphism group of $(\clubsuit)$, $\mathrm{Aut}(\clubsuit)$, as the group of all pairs of permutations (σ, π) with permutation matrices (Σ, Π), such that for all $\boldsymbol{x}$, $J(\boldsymbol{x}) = J(\Pi \boldsymbol{x})$ and $(f_1(\Pi \boldsymbol{x}) \leqslant 0, \ldots, f_m(\Pi \boldsymbol{x}) \leqslant 0) = (f_{\sigma(1)}(\boldsymbol{x}) \leqslant 0, \ldots, f_{\sigma(m)}(\boldsymbol{x}) \leqslant 0)$. In other words, **renaming** the variables yields the same constraints up to reordering. For linear programs (LPs), this is equivalent to $\Sigma A = A\Pi$ and $\Sigma \boldsymbol{b} = \boldsymbol{b}$ and $\boldsymbol{c}^T \Pi = \boldsymbol{c}^T$. The partition that groups together x_i with x_j if some Π in $\mathrm{Aut}(\clubsuit)$ exchanges them is called an **orbit partition**. An interesting fact is that if $\mathcal{P}$ is an orbit partition, $X_{\mathcal{P}}$ is the **symmetrizer matrix** of $\mathrm{Aut}(\clubsuit)$, $X_{\mathcal{P}} = \frac{1}{|\mathrm{Aut}(\clubsuit)|} \sum_{(\Sigma,\Pi) \in \mathrm{Aut}(\clubsuit)} \Pi$. One way to detect renaming symmetries is by inspection of the parameters of the problem. E.g., for a convex quadratic program $(Q, \boldsymbol{c}, A, \boldsymbol{b})$, a set of necessary conditions for the pair of permutations (Σ, Π) to be a renaming symmetry is: (i) $\Pi Q = Q \Pi$ (equivalently $\Pi Q \Pi^T = Q$), (ii) $\boldsymbol{c}^T \Pi = \boldsymbol{c}^T$, (iii) $\Sigma A = A\Pi$, and (iv) $\Sigma \boldsymbol{b} = \boldsymbol{b}$. Such automorphism groups, or rather, the orbit partitions thereof, can be computed via packages such as Saucy (Codenotti et al. 2013). The reason why orbit partitions are lifting partitions of a convex problem, is that $J(X^{\mathcal{P}} \boldsymbol{x}) = J(\frac{1}{|\mathrm{Aut}|} \sum_{(\Sigma,\Pi) \in \mathrm{Aut}} \Pi \boldsymbol{x}) \leqslant \frac{1}{|\mathrm{Aut}|} \sum_{(\Sigma,\Pi) \in \mathrm{Aut}} J(\Pi \boldsymbol{x}) = J(\boldsymbol{x})$, the inequality being due to convexity of J. Reconsider our example on Fig. 2. Permutations renaming row/column 1 to 3 resp. 2 to 4 are automorphisms, and $\mathcal{P}$ is an orbit partition.

For the special case of LPs, (Grohe et al. 2014) have proven that equitable partitions act as lifting partitions. An **equitable partition (EP)** of a square symmetric $n \times n$ matrix M is a partition $\mathcal{P}$ of $1, \ldots, n$, such that $X^{\mathcal{P}}$ satisfies $X^{\mathcal{P}} M = M X^{\mathcal{P}}$. For rectangular matrices, we say that a partition $\mathcal{P}$ of the columns is equitable, if there exists a partition of the rows $\mathcal{Q}$ such that $X^{\mathcal{Q}} M = M X^{\mathcal{P}}$. For LPs, we say that a partition of the variables $\mathcal{P}$ is equitable if there exists a partition of the constraints $\mathcal{Q}$ such that: $\boldsymbol{c}^T X^{\mathcal{P}} = \boldsymbol{c}^T$, $X^{\mathcal{Q}} \boldsymbol{b} = \boldsymbol{b}$, and $X^{\mathcal{Q}} A = A X^{\mathcal{P}}$. EPs and their corresponding partition matrices are referred to as **fractional automorphisms** resp. **fractional symmetries**—we will use both terms in an exchangeable ways—as they satisfy the same conditions as automorphisms from the previous paragraph,

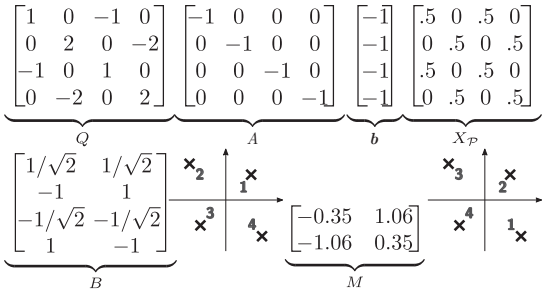

Figure 2: Running example for fractional symmetries of QPs. (Top) A matrix specification of $\mathrm{minimize}_{\boldsymbol{x} \in \mathbb{R}^4} \boldsymbol{x}^T Q \boldsymbol{x}$ s. t. $A\boldsymbol{x} \leqslant \boldsymbol{b}$ and the partition matrix $X^{\mathcal{P}}$ of the partition $\mathcal{P} = \{\{x_1, x_3\}, \{x_2, x_4\}\}$. (Bottom) The factor B with $BB^T = Q$ as well as a sketch of the rows of B. Multiplying B by the matrix M on the right, which equates to rescaling and rotating the vectors by $45°$, is a symmetry of B; it yields the same configuration modulo renaming.

except that $X^{\mathcal{P}}$ is a doubly stochastic matrix and not a permutation matrix. Moreover, EPs have an equivalent combinatorial characterization. A partition $\mathcal{P}$ of $M \in n \times n$ is equitable if for all i, j in the same class P and every class P' (including $P' = P$), we have $\sum_{k \in P'} M_{ik} = \sum_{k \in P'} M_{jk}$. In other words, if we reorder the rows and columns of M such that indices of the same class are next to each other, M will take on a block-rectangular form where every row (and column) of the block has the same sum. One special flavor of EPs are what we will call **counting partitions**, where a narrower condition holds, $|\{k \in P' | M_{ik} = c\}| = |\{k \in P' | M_{jk} = c\}|$ for all $c \in \mathbb{R}$, and $M_{ii} = M_{jj}$ if i, j are in the same class. They partition M into blocks where each row (and column) has the same count of each number. The EP of our example is such a partition. In fact, any orbit partition of a permutation group is a counting partition as well. EPs have several very attractive properties when used as lifting partitions. First, the coarsest WP (as well as the coarsest counting equitable partition) of a matrix is computable in $\mathcal{O}((e + n) \log(n))$ time, where e is the number of non-zeroes in the matrix, via an elegant algorithm called color refinement(Grohe et al. 2014). Second, the coarsest EP is at least as coarse as the orbit partition of a matrix, hence it offers more compression.

Fractional Symmetry of Convex QPs

Having developed automorphisms of convex QPs, we now move on to our main contributions. We develop FA esp. EPs of a convex QP. We start off with showing that they are lifted partitions. Then, we provide a geometric interpretation and investigate whether kernels preserve fractional symmetries.

Equitable Partitions (EPs) of QPs: We prove now that the lifting partition of a convex QP captures its symmetries.

Theorem 1. *Let $\boldsymbol{QP} = (Q, \boldsymbol{c}, A, \boldsymbol{b})$ be a convex quadratic program. If $\mathcal{P}$ is a partition of the variables of $\boldsymbol{QP}$, such that: (a) $X^{\mathcal{P}} Q = Q X^{\mathcal{P}}$ and $\boldsymbol{c}^T X^{\mathcal{P}} = \boldsymbol{c}^T$, (b) there exists a partition $\mathcal{Q}$ of the constraints of $\boldsymbol{QP}$ such that $X^{\mathcal{Q}} \boldsymbol{b} = \boldsymbol{b}$ and $X^{\mathcal{Q}} A = A X^{\mathcal{P}}$, then $\mathcal{P}$ is a lifting partition for $\boldsymbol{QP}$.*

Proof. We proceed along the lines drawn out in the previous section and show that for any feasible $\boldsymbol{x}$, $\boldsymbol{x}' = X^{\mathcal{P}}\boldsymbol{x}$, the class-averaged $\boldsymbol{x}$, is both feasible and $J(\boldsymbol{x}') \leqslant J(\boldsymbol{x})$. Let us start with the latter. Note that both Q and $\boldsymbol{X}^{\mathcal{P}}$ are diagonalizable (i.e. admit an eigendecomposition) since they are symmetric and real matrices. It is known that if two diagonalizable matrices commute (as is our starting hypothesis, $X^{\mathcal{P}}Q = QX^{\mathcal{P}}$), then they are also simultaneously diagonalizable. That is, there exists an orthonormal basis of vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ such that $Q = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T = U\Lambda U^T$ and $X^{\mathcal{P}} = \sum_i \kappa_i \boldsymbol{u}_i \boldsymbol{u}_i^T = UKU^T$, where the λ_i's and κ_i's are nonnegative scalars. Now, $J(\boldsymbol{x}') = J(X^{\mathcal{P}}\boldsymbol{x}) = \boldsymbol{x}^T(X^{\mathcal{P}})^T Q X^{\mathcal{P}}\boldsymbol{x} + \boldsymbol{c}^T X^{\mathcal{P}}\boldsymbol{x}$. From our discussion so far and assumption (a), this is equal to $\boldsymbol{x}^T UK^T U^T U\Lambda U^T UKU^T\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{x}^T U\Lambda K^2 U^T\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$. The key observation is that because $X^{\mathcal{P}}$ is doubly stochastic, $|\kappa_i| \leqslant 1$. Hence $\boldsymbol{x}^T UK^2\Lambda U^T\boldsymbol{x} = \sum_i \kappa_i^2 \lambda^i \boldsymbol{x}^T\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{x} \leqslant \sum_i \lambda^i \boldsymbol{x}^T\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{x}$ as $\lambda_i \boldsymbol{x}^T\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{x}$ is a nonnegative quantity. This entails $J(\boldsymbol{x}) \geqslant J(\boldsymbol{x}')$.

Regarding feasibility, because $X^{\mathcal{Q}}$ is a matrix of nonnegative numbers, $A\boldsymbol{x} \leqslant \boldsymbol{b}$ implies $X^{\mathcal{Q}}A\boldsymbol{x} \leqslant X^{\mathcal{Q}}\boldsymbol{b}$. Due to (b), this becomes $AX^{\mathcal{P}}\boldsymbol{x} \leqslant \boldsymbol{b}$, that is, $A\boldsymbol{x}' \leqslant \boldsymbol{b}$, demonstrating the feasibility of $\boldsymbol{x}'$. We have thus satisfied the two sufficient conditions stated in the previous section and shown that any $\mathcal{P}$ satisfying our assumptions is a lifting partition for $\boldsymbol{QP}$. □

Example. Recall Q from our running example on Fig. 2. However, this time we propose $\mathcal{P}' = \{\{x_1, x_2, x_3, x_4\}\}$ as a lifting partition with $X^{\mathcal{P}'} = \frac{1}{4}\cdot 1_4$, where 1_4 is the 4×4 matrix of ones. We observe that $QX^{\mathcal{P}'} = X^{\mathcal{P}'}Q = 0_4$, moreover, if we introduce the constraint partition $\mathcal{Q}' = \{y_1, ..., y_4\}$ with partition matrix $X^{\mathcal{Q}'} = X^{\mathcal{P}'}$, we have that $X^{\mathcal{Q}'}A = AX^{\mathcal{P}'}$ and $X^{\mathcal{Q}'}\boldsymbol{b} = \boldsymbol{b}$. According to Thm. 1 $\mathcal{P}'$ is a lifting partition of the QP in question. □

There are two interesting observations to be made here. First, we have gained even further compression over our previous attempt, having a compressed problem with 1 variable instead of 2. Second, there is no automorphism of Q that could possibly exchange x_1 and x_2. As fractional symmetries generalize exact symmetries, see e.g. (Godsil 1997), it is to be expected that coarser equitable partitions than the orbit partition Q could satisfy the conditions of Thm 1. Moreover, these observations allow one to gain insight into what fractional symmetry means geometrically for a dataset. This is important as the matrix Q relates to the data we feed into the optimization problem for many QPs; e.g., in the SVM dual QP, the entries of Q are inner products of the training feature vectors.

Geometry of Fractionally-Symmetric QPs: Our investigation is inspired by the characterization of automorphisms of semidefinite matrices and quadratic forms. One way to think about a semidefinite matrix Q is as the Gram matrix of a set of vectors, i.e. $Q = BB^T$ where B is an $n\times k$ matrix and $k \geqslant \operatorname{rank}(Q)$. In this light, the quadratic form $\boldsymbol{x}^T Q\boldsymbol{x}$ can be seen as the squared Euclidean norm of a matrix-vector product. That is, $\boldsymbol{x}^T Q\boldsymbol{x} = \boldsymbol{x}^T BB^T\boldsymbol{x} = (B^T\boldsymbol{x})^T(B^T\boldsymbol{x}) = ||B^T\boldsymbol{x}||^2$. It is a basic fact that the Euclidean norm is invariant under orthonormal transformations, that is, for any orthonormal matrix O and any vector $\boldsymbol{y}$, $||O^T\boldsymbol{y}|| = \boldsymbol{y}$ as $\boldsymbol{y}^T OO^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{y}$. Thus, suppose we have a **rotational automorphism** of B, i.e., a pair of orthonormal matrix O and permutation matrix Π, such that $\Pi B = BO$ or also $\Pi BO^T = B$. That is, rotating the tuple of vectors that are the rows of B together yields same tuple back, but in different order. Observe then, that Π would be a renaming automorphism for Q, since $\Pi Q\Pi^T = \Pi BO^T OB\Pi^T = BB^T = Q$, implying $\Pi Q = Q\Pi$. Moreover, if the right dimension (number of columns) B is held fixed, the converse is true as well (Bremner, Dutour Sikrić, and Schürmann 2009). That is, not only do rotational symmetries of B correspond to renaming symmetries of Q, but vice-versa, as for fixed k, the semidefinite factors of Q are unique up to rotations.

Example. Our Q from Fig. 2 can be factored into BB^T as shown on Fig. 2. The Figure also shows the plot of these vectors. If we were to rotate them by 180° counter-clockwise, we would get back the same set of vectors, but in the order $\{x_3, x_4, x_1, x_2\}$. The permutation matrix according to this reordering is a renaming automorphism of Q. □

Using the case of automorphisms as a motivation, we now turn to fractional automorphisms. More precisely, given a doubly stochastic and idempotent matrix X, such that $XQ = QX$, we would like to derive a similar characterization of X in terms of B. As we prove now, this is indeed possible.

Theorem 2. *Let X be a symmetric and idempotent (as our usual color-refinement automorphisms are) matrix, and $Q = BB^T$ be a positive semidefinite matrix with B having full column rank. Then $XQ = QX$* if and only if *there exists a symmetric matrix R such that $XB = BR$.*

Proof. (**only if direction**): Let R be such that $R = R^T$ and $XB = BR$. Then, $XQ = XBB^T = BRB^T$. Making use of $R = R^T$ this rewrites as $BR^T B^T = B(BR)^T = B(XB)^T = BB^T X^T = QX$, as X is symmetric.

(**if direction**): Let $XQ = QX$ with X being idempotent and symmetric. Then, let $R = B^T XB(B^T B)^{-1}$. Observe that $B(B^T B)^{-1}$ exists and is the right pseudoinverse of $B^T, i.e., B^T B(B^T B)^{-1} = I_k$, as B has full column rank. Therefore, left multiplying by I_k yields $XB = XBB^T B(B^T B)^{-1} = BB^T XB(B^T B)^{-1} = BR$. It remains to demonstrate that R is symmetric. Recall that $R^T R$ and $(R^T R)^{-1}$ are symmetric matrices. Then, $R^T R = \left[B^T XB(B^T B)^{-1}\right]^T B^T XB(B^T B)^{-1}$. Since, $(B^T B)^{-T} = (B^T B)^{-1}$ and $XBB^T = BB^T X$, this simplifies to $(B^T B)^{-1}(B^T B)XXB(B^T B)^{-1}$. Since $XX = X$ and using I_k, this simplifies to $B^T XB(B^T B)^{-1} = R$. Hence, as $R^T R = R$, R is symmetric. □

This theorem holds the key to explaining why all 4 dimensions in our example are compressed together. To see why, consider the situation on Fig 2.

Example. Fig. 2 shows the factor B of Q (as well as a sketch of its rows) and an invertible matrix M, which consists of a clockwise rotation by 45° which aligns the vectors with the axes, a rescaling of the vectors along the axes, then a further 45°. Multiplying B by M yields back the same row vectors modulo a cyclic permutation, exchanging x_4 with x_1, x_1 with x_2 and so on, i.e. $\Sigma B = BM$. Moreover $BMM^T B^T \neq Q$. The group of $\{M, M^2, M^3, M^4\}$ is thus a group that does not correspond to any group of automorphisms of Q, yet, the symmetrizer matrix $\frac{1}{4}\sum_{i=1}^4 M^i$ is symmetric (and equal to

0_2), so it qualifies under the conditions of Thm. 2. □

From this we can conclude that certain scaling symmetries of B do not result in symmetries of Q, but do result in **fractional symmetries** of Q (Thm. 2). On the other hand, by Thm. 1, we can also infer that these symmetries can safely be compressed out when minimizing the quadratic form $\boldsymbol{x}^T Q \boldsymbol{x}$. Note finally that even these symmetries do not exhaust the possible matrices of Thm. 2, unless the graph isomorphism problem is P-TIME solvable. Thm. 2 allows for partitions and matrices that do not correspond to any group. Characterizing them is an exciting avenue for future work.

Approximately Lifted SVMs: Indeed, one may argue that the (rotational) automorphism group of most Euclidean datasets consists of the identity transformation alone. This follows from the same result for convex bodies, see e.g. (Güler and Gürtuna 2012), and is to be expected, since the symmetry properties of a given dataset B can easily be destroyed by slightly perturbing the body. To bypass this, we propose the first approximate lifting approach for Euclidean datasets.

Proposition 3. *Let B be an Euclidean dataset and D its corresponding pairwise distance matrix. Then $B_{i\cdot}$ and $B_{j\cdot}$ are in the same (rotational) orbit if an only if $B_{i\cdot}$ and $B_{j\cdot}$ have the same sorted distances to all other data points.*

Proof. The EP of D encodes the symmetries of B. To compute it, we represent it as a colored graph C of D. We note that C is a clique with edge colors encoding distances. We turn this into a node-colored graph by assigning the same color to all nodes that have identical edge-color signatures. Running color-refinement on this graph does not add any new color since C is a clique. □

This suggest a simple way to compute proper approximations of (rotational) EPs of B as illustrated in Fig. 3(left): (1, optional) Whiten the data to capture some scalings, (2) compute the pairwise distance matrix D of B (potentially using anchor points), (3) sort each row of D, and (4) run any cluster algorithm on the sorted distance matrix. More importantly, it connects lifted inference to SVM approaches that reduce the size of the optimization problem by extracting a small number of representatives from the original training set and using them to train an approximate SVM, see e.g. (Boley and Cao 2004). In particular, one can easily prove (proof left out due to space restrictions) that the same PAC-style generalization bound applies (Cao and Boley 2007): the approximately lifted SVM will very likely have a small expected error rate if it has a small empirical loss over the original dataset.

Kernels and Equitable Partitions: Finally, we touch upon the relationship between the fractional symmetry of data vectors and kernels. Kernel functions often appear in conjunction with quadratic optimization in machine learning problems as a means of enriching the hypothesis space of a learner. From an algebraic perspective, the essence of the approach is to replace the entries of the semidefinite matrix Q with the values of a kernel function, which represents the inner product of data vectors under some non-linear transformation in a high dimensional space. That is, in place of $Q_{ij} = \langle B_{\cdot i}, B_{\cdot j} \rangle$, we use $K_{ij} = k(B_{\cdot i}, B_{\cdot j}) = \langle \phi(B_{i\cdot}), \phi(B_{j\cdot}) \rangle$, where $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is some non-linear function with m much greater than n or even infinite. Due to the prevalence of kernels, it is important to understand whether kernels preserve or destroy symmetries. Here, we will examine two popular kernels, the polynomial kernel, $k^{\mathrm{POLY}}(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 1)^g$ and $k^{\mathrm{RBF}}(\boldsymbol{x}, \boldsymbol{y}) = \exp(-2\gamma^2 ||\boldsymbol{x} - \boldsymbol{y}||_2^2)$, where g is a positive integer and γ is a nonzero real number. We find that in both cases, if $Q = BB^T$ admits a **counting equitable partition**, then K will admit the same partition as well, i.e., these two kernels preserve fractional symmetry of Q up to counting (recall, that includes rotational symmetry of B):

Proposition 4. *Let B be a matrix whose rows are data instances. Then, if $Q = BB^T$ admits a counting equitable partition $\mathcal{P}$ with partition matrix $X^{\mathcal{P}}$, then both kernel matrices* **(a)** K^{POLY} *and* **(b)** K^{RBF} *of this set of vectors admit the same counting partition.*

Proof. Recall that an EP $\mathcal{P}$ is a counting partition for Q if for all x_i, x_j in the same class P, and for every class P' (including $P' = P$), $|\{x_k \in P' | Q_{ik} = c\}| = |\{x_k \in P' | Q_{jk} = c\}|$ for all $c \in \mathbb{R}$, and $Q_{ii} = Q_{jj}$. **(a)** A direct consequence of this definition is that if $\mathcal{P}$ is a counting partition for Q, it will be a counting partition for every other matrix whose equality pattern respects that of Q, in other words, $Q_{ij} = Q_{pq} \Rightarrow K_{ij} = K_{pq}$. K^{POLY} has exactly this property: $K_{ij}^{\mathrm{POLY}} = (\langle B_{i\cdot}, B_{j\cdot} \rangle + 1)^g = (Q_{ij} + 1)^g$. It is clear that if $Q_i j$ and $Q_p q$ are equal, the values of the last expression would be equal as well. **(b)** First, we note $K_{ij}^{\mathrm{RBF}} = \exp(-2\gamma^2 ||B_{i\cdot}||^2) \exp(-2\gamma^2 ||B_{j\cdot}||^2) \exp(-\gamma^2 \langle B_{i\cdot}, B_{j\cdot} \rangle)$. This allows one to rewrite K^{RBF} in terms of Q: $K_{ij}^{\mathrm{RBF}} = \exp(-2\gamma^2 Q_{ii}) \exp(-2\gamma^2 Q_{jj}) \exp(-\gamma^2 Q_{ij})$. Now, let $x_i, x_j \in P$ and $x_p, x_q \in P'$ such that $Q_{ip} = Q_{jq}$. Since $Q_{ii} = Q_{jj}$ (by virtue of being in P) and $Q_{pp} = Q_{qq}$ (by virtue of P'), we have that $K_{ip}^{\mathrm{RBF}} = K_{jq}^{\mathrm{RBF}}$ hence counts across classes are preserved. □

To summarize, convex QP can be lifted without loss of quality: compute in quasi-linear time its quotient model w.r.t its EP as illustrated in Fig. 3(right). For the polynomial and RBF kernels, this also leads to valid liftings.

Empirical Illustration

Our intention here is to investigate whether **(Q)** AI can potentially benefit from relational and lifted QPs?

As our main experiment, we compared our lifted QP approach to relational linear programs (Kersting, Mladenov, and Tokmakov 2015), following their experimental setup for the Cora dataset (Sen et al. 2008) consisting of 2708 scientific papers classified into seven classes. Each paper is described by a binary word vector indicating the absence/presence of a word from a dictionary of 1433 words. The citation network of the papers consisting of 5429 links. The goal is to predict the class of the paper. For simplicity, we converted this problem to a binary classification problem by taking the largest of the 7 classes as a positive class. We compared four different learners on Cora. The base classifiers are an ∞-norm regularized SVM (LP-SVM) (Zhou, Zhang, and Jiao 2002) and a conventional SVM (QP-SVM) (Vapnik 1998) formulated as a convex QP. Both use the word feature vectors and do standard linear prediction (no kernel used). Additionally we considered transductive, collective versions of both of them, again

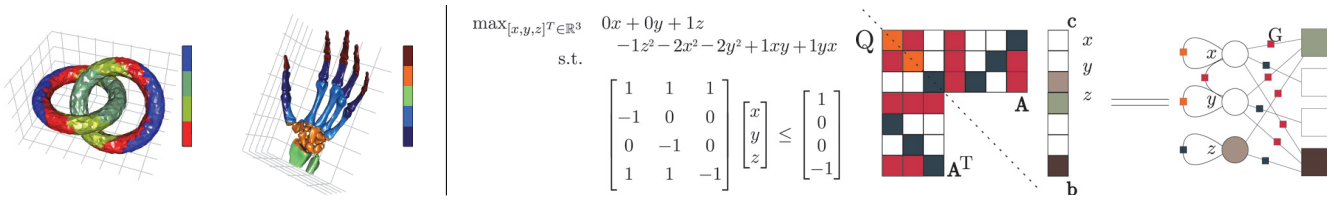

Figure 3: (Left) Approximate EPs without scalings on 3D datasets. The colors encode the rotational symmetries under a budget of 4 resp. 5 orbits. E.g. the"hand" consists of 327.323 points (a clique with $50 \cdot 10^{10}$ edges) running in < 5 secs using 2500 anchor points. (Right) Convex QPs can be exactly lifted by running color-refinement (Ahmadi et al. 2013) on the colored graph encoding it. This takes quasi-linear time, and one can directly read off $X^{\mathcal{P}}$. After color-refinement, nodes with the same color form the quotient QP of the original QP. Since the corresponding constraints of variables with the same color (they are in the same orbit) are identical, one can drop all but one of the identical constraints; this forms $X^{\mathcal{Q}}$. This compression is exact, i.e., the lifted QP computes an optimal solution of the original QP. (Best viewed in color)

following (Kersting, Mladenov, and Tokmakov 2015), denoted as TC-LP-SVM resp. TC-QP-SVM. Both transductive approaches have access to the citation network and implement the following simple rule: whenever we have access to an unlabeled paper i, if there is a cited or citing labeled paper j, then assume the label of j as a label of i. To account for contradicting constraint (a paper citing both papers of and not of its class), we introduced separate slack variables for the transductive constraints and add them to the objective with a different penalty parameter. This can easily be implemented by adding a few relational constraints to an existing standard QP-SVM formulation, see Fig. 1(left). In order to investigate the performance, we varied the amount of labeled examples available. That is, we have four cases, where we restricted the number of labeled examples to $t = 20\%, 40\%, 60\%$, and 80% of size of the dataset. We first randomly split the dataset into a labeled set L and an unlabeled test set B, according to t. Then, we split L randomly in half, leaving one half A for training, the other half becoming a validation set C. The validation set was used to select the parameters of the TC-QP-SVM in a 5-fold cross-validation fashion. That is, we split the validation set into 5 subsets C_i of equal size. On these sets we selected the parameter using a grid search for each C_i on a $A \cup (C \backslash C_i)$ labeled and $B \cup C_i$ unlabeled examples, computing the prediction error on C_i and averaging it over all C_is. We then evaluated the selected parameters on the test set B whose labels were never revealed in training. We repeated this experiment 5 times (one for each C_i) for the TC-SVMs. For consistency, we followed the same protocol with QP-SVM and LP-SVM, except that the set $B \cup C_i$ did not appear during training as the non-transductive learners have no use for unlabeled examples. That is, we selected parameters by training on $A \cup (C \backslash C_i)$ and evaluating on C_i. The selected parameters were then evaluated on the test set B. For all SVM models, we also ran a ground and a lifted version. The results are summarized in Fig. 1(right): QP-SVM outperforms LP-SVM for each setting, both are outperformed by TC-L/QP-SVM, and TC-QP-SVM outperforms TC-LP-SVM. While there was no appreciable symmetry in either QP-SVM or LP-SVM, TC-QP-SVM exhibited significant variable and constraint reduction: the lifted problem was reduced to up to 78% of the variables, resp., 70% of the constraints of the ground problem, while achieving the same accuracy[1].

To illustrate our approach on propositional data, we considered SVM classifiers for varying amounts of overlap between two classes represented by spherical Gaussians. This dataset was chosen in order to depict the potential of approximate symmetries. We trained a lifted SVM (LSVM) with 200 approximate color classes and a conventional SVM, both with RBF kernels, on 2500 training examples per class. We used a grid search together with CV for selecting $\gamma = \{0.25, 0.50, 1.00, 2.00, 4.00\}$ and $C = \{0.5, 1.0, 2.0\}$. The performance was measured on an independently drawn test set of 5000 data points per class. For approximate lifting we used k-Means using the Euclidean metric and 500 anchor points. For 4 units apart class centers, the SVM achieved an error of 0.02 in 20 secs (all numbers in this experiment are averaged over 10 reruns and rounded to the second digit), while the LSVM achieved 0.02 in 1.7 secs. For 2 units apart class centers, the SVM took 98 secs achieving an error of 0.16, while the LSVM achieved 0.17 in 2.1 secs. Thus, the generalization bound guarantees that the LSVM will very likely have an expected error rate comparable to the SVM, at a fraction of time.

Overall, the empirical results are clear evidence for an affirmative answer to question **(Q)**.

Conclusions

We have deepened the understanding of symmetries in statistical AI and extended the scope of lifted inference. Specifically, we have introduced and studied a precise mathematical definition of fractional symmetry of convex QPs. Using the tool of fractional automorphism, orbits of optimization variables are obtained, and lifted solvers materialize as performing the corresponding optimization problem in the space of per-orbit optimization variables. This enables the lifting of a large class of AI tools such as spectral relaxations for MRF inference (Cour and Shi 2007) and quadratic assignments problems as implied by (Lu and Boutilier 2015). We here instantiated the framework for QP-SVMs by developing the first lifted and relational solvers for them and illustrating

[1]Qualitatively similar results were obtained on the two-moons dataset with 150 additional features, each drawn randomly from a Gaussian per example, using the 4-NN graph as "citation network".

empirically their benefits. In the future, other AI settings and more datasets should be explored, one should investigate the link to other data reduction methods, move beyond convex QPs, and explore our framework for symmetry-based learning (Gens and Domingos 2014)[2].

Acknowledgments: The authors would like to thank the anonymous reviewers for their feedback. The work was partly supported by the DFG Collaborative Research Center SFB 876, project A6 "Resource-efficient Graph Mining".

References

Ahmadi, B.; Kersting, K.; Mladenov, M.; and Natarajan, S. 2013. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning* 92:91–132.

Boley, D., and Cao, D. 2004. Training support vector machines using adaptive clustering. In *Proceedings of the 4th SIAM International Conference on Data Mining (SDM)*, 126–137.

Bremner, D.; Dutour Sikrić, M.; and Schürmann, A. 2009. Polyhedral representation conversion up to symmetries. In *Proceedings of the 2006 CRM Workshop on Polyhedral Computations*. AMS, Providence.

Bui, H.; Huynh, T.; and Riedel, S. 2013. Automorphism groups of graphical models and lifted variational inference. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Cao, D., and Boley, D. 2007. A PAC bound for approximate support vector machines. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, 455–460.

Codenotti, P.; Katebi, H.; Sakallah, K.; and Markov, I. 2013. Conflict analysis and branching heuristics in the search for graph automorphisms. In *Proc. of ICATI*, 907–914.

Cour, T., and Shi, J. 2007. Solving markov random fields with spectral relaxation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 75–82.

De Raedt, L.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Morgan & Claypool Publishers.

Diamond, S.; Chu, E.; and Boyd, S. 2014. CVXPY: A Python-embedded modeling language for convex optimization, version 0.2. http://cvxpy.org/.

Gens, R., and Domingos, P. 2014. Deep symmetry networks. In *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2537–2545.

Godsil, C. 1997. Compact graphs and equitable partitions. *Linear Algebra Applications* 255:259–266.

[2]Initial results are promising. We considered MNIST images for two randomly selected classes and augmented the dataset by applying rotations $(1^\circ, 2^\circ, \ldots, 360^\circ)$ to each images. An SVM learned on all (original+augmented) examples achieved 95.9% accuracy on the independent test set, taking 10.5 hours. In contrast, an LSVM with assumed symmetries—putting rotations of $(0^\circ, 1^\circ, \ldots, 9^\circ), (10^\circ, \ldots, 19^\circ), \ldots$ degrees into the same color class—achieved 96.0% accuracy in less then 5 minutes. For both classifiers, cost parameters were selected using a grid-search on a subset of the training set.

Grohe, M.; Kersting, K.; Mladenov, M.; and Selman, E. 2014. Dimension reduction via colour refinement. In *Proceedings of the 22th Annual European Symposium on Algorithms (ESA)*, 505–516.

Güler, O., and Gürtuna, F. 2012. Symmetry of convex sets and its applications to the extremal ellipsoids of convex bodies. *Optimization Methods and Software* 27(4-5):735–759.

Jernite, Y.; Rush, S.; and Sontag, D. 2015. A fast variational approach for learning markov random field language models. In *Proc. of the 32nd International Conference on Machine Learning (ICML)*.

Kersting, K.; Mladenov, M.; and Tokmakov, P. 2015. Relational linear programming. *Artificial Intelligence Journal (AIJ)* OnlineFirst.

Lu, T., and Boutilier, C. 2015. Value-directed compression of large-scale assignment problems. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*.

Margot, F. 2010. Symmetry in integer linear programming. In Jünger, M.; Liebling, T.; Naddef, D.; Nemhauser, G.; Pulleyblank, W.; Reinelt, G.; Rinaldi, G.; and Wolsey, L., eds., *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer. 1–40.

Mladenov, M.; Globerson, A.; and Kersting, K. 2014. Lifted message passing as reparametrization of graphical models. In *Proc. of the 30th Int. Conf. on Uncertainty in Artificial Intelligence (UAI)*.

Poole, D. 2003. First-order probabilistic inference. In *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 985–991.

Ré, C.; Agrawal, D.; Balazinska, M.; Cafarella, M.; Jordan, M.; Kraska, T.; and Ramakrishnan, R. 2015. Machine learning and databases: The sound of things to come or a cacophony of hype? In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 283–284.

Riedel, S.; Smith, D.; and McCallum, A. 2012. Parse, Price and Cut–Delayed Column and Row Generation for Graph Based Parsers. In *Proc. of the EMNLP-CoNLL*, 732–743.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI Magazine* 29(3):93–106.

Shervashidze, N., and Borgwardt, K. 2009. Fast subtree kernels on graphs. In *Proc. of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, 1660–1668.

Vapnik, V. 1998. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications and control series. John Wiley & Sons, New York. A Wiley-Interscience Publication.

Wallace, S., and Ziemba, W., eds. 2005. *Applications of Stochastic Programming*. SIAM, Philadelphia.

Yih, W., and Roth, D. 2007. Global inference for entity and relation identification via a linear programming formulation. In Getoor, L., and Taskar, B., eds., *An Introduction to Statistical Relational Learning*. MIT Press.

Zhou, W.; Zhang, L.; and Jiao, L. 2002. Linear programming support vector machines. *Pattern recognition* 35(12):2927–2936.