Carnegie Mellon University
Research Showcase @ CMU

Machine Learning Department | School of Computer Science

12-2013

A projection algorithm for strictly monotone linear complementarity problems

Erik Zawadski
Carnegie Mellon University

Geoffrey J. Gordon
Carnegie Mellon University, ggordon@cs.cmu.edu

Andre Platzer
Carnegie Mellon University, aplatzer@cs.cmu.edu

Follow this and additional works at: http://repository.cmu.edu/machine_learning

 Part of the Theory and Algorithms Commons

Published In
Proceedings of NIPS OPT2013: Optimization for Machine Learning.

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Machine Learning Department by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

A projection algorithm for strictly monotone linear complementarity problems.*

Erik Zawadzki
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
epz@cs.cmu.edu

Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

André Platzer
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
aplatzer@cs.cmu.edu

Abstract

Complementary problems play a central role in equilibrium finding, physical simulation, and optimization. As a consequence, we are interested in understanding how to solve these problems quickly, and this often involves approximation. In this paper we present a method for approximately solving strictly monotone linear complementarity problems with a Galerkin approximation. We also give bounds for the approximate error, and prove novel bounds on perturbation error. These perturbation bounds suggest that a Galerkin approximation may be much less sensitive to noise than the original LCP.

1 Introduction

Complementarity problems arise in a wide variety of areas, including machine learning, planning, game theory, and physical simulation. In particular, both the problem of planning in a belief-space Markov decision processes (MDPs) and training support vector machines (SVMs) are (large) monotone linear complementarity problems (LCPs). In all of the above areas, to handle large-scale problem instances, we need fast approximate solution methods. For example, for belief-space MDPs, this requirement translates to fast approximate planners.

One promising idea for fast computation is Galerkin approximation, in which we search for the best answer within the span of a given set of basis functions. Solutions are known for a number of special cases of this problem with restricted basis sets, but the general-basis problem (even just for MDPs) has been open for at least 50 years, since the work of Richard Bellman in the late 1950s.

We have made four concrete steps toward a solution to the problem of Galerkin approximation for LCPs and variational inequalities: a new solution concept, perturbation bounds for this concept, and two new algorithms which compute solutions according to this concept. These algorithms are only proven to have low approximation error for *strictly* monotone LCPs (a class which does not include MDPs, although for many MDPs we can easily "regularize" into this class). One of the algorithms is guaranteed to converge for any (strict or non-strict) monotone LCP, although we do not yet have error bounds. We are currently working to extend both the algorithms and proofs.

The first algorithm is similar to projected gradient descent. It is related to recent work by Bertsekas [1], who proposed one possible Galerkin method for variational inequalities. However, the Bertsekas method can exhibit two problems in practice: its approximation error is worse than might be expected based on the ability of the basis to represent the desired solution, and each iteration requires a projection step that is not always easy to implement efficiently.

*This material is based upon work supported by the Army Research Office under Award No. W911NF-09-1-0273 and by ONR MURI grant number N00014-09-1-1052. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Army Research Office.

To resolve these problems, we present a new Galerkin method with improved behavior: the new error bounds depend directly on the distance from the true solution to the subspace spanned by a basis, and the only projections required are onto the feasible region or onto the span of a basis. The new Galerkin method also gives us a way to analyze how perturbation affects the solution of an LCP. We prove that the Galerkin approximation makes the Galerkin LCP solution less sensitive to noise.

The new solution concept can be defined as the fixed point of the above algorithm. One can interpret the new solution concept as a modified LCP, which has a particular special structure: the LCP matrix $M \in \mathbb{R}^{n\times n}$ can be decomposed as $M = \Phi U + \Pi_0$, where the rank of Φ is $k < n$, and Π_0 denotes Euclidean projection onto the nullspace of $\Phi^\top$. Such LCPs are called *projective*.

Given this solution concept, we can ask if there are other algorithms to achieve it, with different or better convergence properties. It turns out that an interior-point method can be highly effective: its convergence is exponentially faster than gradient descent, but each iteration requires solving a system of linear equations, so it is limited to problems whose size or structure lets us solve this system efficiently. In addition, the method provably solves (strict or non-strict) monotone projective LCPs.

In particular, our paper [5] gives an algorithm that solves a monotone projective LCP to relative accuracy ϵ in $O(\sqrt{n}\ln(1/\epsilon))$ iterations, with each iteration requiring $O(nk^2)$ flops. This complexity compares favorably with interior-point algorithms for general monotone LCPs: these algorithms also require $O(\sqrt{n}\ln(1/\epsilon))$ iterations, but each iteration needs to solve an $n \times n$ system of linear equations, a much higher cost than our algorithm when $k \ll n$. This algorithm works even though the solution to a projective LCP is not restricted to lie in the low-rank subspace spanned by Φ.

2 Complementarity problems

A *complementarity problem* (CP) (see, *e.g.*, [4]) is defined by a function $F : \mathbb{R}^n \to \mathbb{R}^n$ and a cone K. A solution to a CP is any vector x such that $K \ni x \perp F(x) \in K^*$, where $\perp$ denotes orthogonality and K^* is the dual cone of K—$K^* = \{y \in \mathbb{R}^n \mid \forall x \in K, y^\top x \geq 0\}$.

A *linear CP* (LCP) is a restriction of the more general CP to affine functions $F(x) = Mx + q$ and $K = [0,\infty)^n \equiv \mathbb{R}^n_+$ (see, *e.g.*,[3]). Since the cone is fixed, we can unambiguously describe an LCP instance with the pair (M, q). LCPs can express the Karush-Kuhn-Tucker (KKT) system (see, *e.g.*, [2]) for any quadratic program (QP), including non-convex QPs. This is a broad class of problems—SAT can be reduced in polynomial time to an LCP. We restrict our attention to a class of LCPs with unique solutions that can be solved in polynomial time: strictly monotone LCPs.

Def. 1. *A function $F : \mathbb{R}^n \to \mathbb{R}^n$ is β-monotone for $\beta \geq 0$ on a set $C \subseteq \mathbb{R}^n$ if*

$$\forall x, y \in C,\ (x-y)^T(F(x) - F(y)) \geq \beta\|x-y\|^2. \tag{1}$$

A CP is monotone if F is β-monotone on K for $\beta \geq 0$. We say that the CP is strictly monotone if F is β-monotone for $\beta > 0$. Strictly monotone CPs always have a unique solution x^* [3]. An LCP (M, q) is monotone iff M is positive semidefinite, and is strictly monotone iff M is positive definite.

A strictly monotone CP can be solved in polynomial time with an iterative projection method:

$$x^{(t+1)} = \Pi_K\left[x^{(t)} + \alpha F\left(x^{(t)}\right)\right]. \tag{2}$$

Here, Π_K is the orthogonal projection onto K. For LCPs this is component-wise thresholding $\Pi_{\mathbb{R}^n_+}(x) \equiv x_+ = \max(0, x)$. $\alpha > 0$ is the 'step size' or 'learning rate'.

This projection operator T is a contraction with a Lipschitz constant based on the strong monotonicity constant β and Lipschitz constant L for F. For a strictly monotone LCPs and the Euclidian norm, $L = \sigma_{max}(M) \geq \beta = \sigma_{min}(M) > 0$.

Lem. 1 (Lemma 1.1 and 2.2 in [6])**.** *If F is strictly monotone with coefficient β and Lipschitz with constant L and if $\alpha = \beta/L^2$, then T is a contraction with Lipschitz constant $\gamma = \sqrt{1-\beta^2/L^2} < 1$.*

For proofs of these results, see our tech report [6]. Consequently, to ensure that an iteration $x^{(t)}$ is within an $\epsilon > 0$ ball of the true solution x^*—*i.e.* $\|x^{(t)} - x^*\| \leq \epsilon$—we can repeat the projection procedure (2) until $t \geq \ln(\epsilon)/\ln(\gamma)$.

2.1 Galerkin methods

If n is large, then we must use an appropriate approximate solver. One way to approximate the above projection algorithm is to look for approximate solution within the span of a basis matrix Φ. We do this by adding a step to (2) that projects onto $\mathsf{span}(\Phi)$ with the orthogonal projection operator Π_Φ:

$$z^{(t)} = \Pi_\Phi \left[x^{(t)} - \alpha F(x^{(t)} \right], \qquad x^{(t+1)} = \Pi_K(z^{(t)}) \tag{3}$$

More details of this methods can be found in our tech report [6].

By contrast, Bertsekas' Galerkin projection method [1] projects onto $\hat{K} = \mathsf{span}(\Phi) \cap K$ which has two disadvantages. First, it is harder to project onto $\hat{K}$ than to project onto $\mathsf{span}(\Phi)$ first, and then onto K. For LCPs, projection onto the span is a rank k linear operator that may be precomputed via the QR decomposition, and projection onto $\mathbb{R}^n_+$ is component-wise thresholding; denoted by $[\cdot]_+$. Second, the distance between the fixed point for Bertsekas-Galerkin and the true solution depends on $\|x^* - \Pi_{\hat{K}} x^*\|$ (see, *e.g.*, Lemma 3.2 in [6]). Our algorithm depends only on $\|x^* - \Pi_\Phi x^*\|$, which is no larger than $\|x^* - \Pi_{\hat{K}} x^*\|$ and may be much smaller. This is because $\hat{K} \subseteq \mathsf{span}(\Phi)$ so for any arbitrary point x, the points in $\mathsf{span}(\Phi)$ are no further than the points in $\hat{K}$.

The Galerkin update (3), viewed from either x and z's perspective, is a contraction for any strongly monotone LCP.

Lem. 2 (Lemma 4.1 in [6]). *If $(I - \alpha F)$ is γ-Lipschitz, then the Galerkin update (3) is γ-Lipschitz for both x and z.*

This follows from projections being 1-Lipschitz; composing projection with a contraction yields a contraction. As a consequence the Galerkin update has a unique fixed point $(\bar{x}, \bar{z})$ that it will be within $\|x^{(t)} - \bar{x}\| \leq \epsilon \|x^{(0)} - \bar{x}\|$ after $t \geq \ln \epsilon / \ln \gamma$ iterations.

This fixed point $(\bar{x}, \bar{z})$ is not that far from the true solution x^*.

Lem. 3 (Lemma 4.2 in [6]). *The error bound between the Galerkin solution from (3) and x^* is bounded:*

$$\|\bar{x} - x^*\| \leq \|x^* - \Pi_\Phi(x^*)\| / (1 - \gamma). \tag{4}$$

Similarly:

$$\|\bar{z} - x^*\| \leq \|x^* - \Pi_\Phi(x^*)\| / (1 - \gamma). \tag{5}$$

This lemma follows from the representation error being contracted by γ after every iteration.

The Galerkin update can be implemented efficiently for LCPs since the projection onto a subspace spanned by Φ can be done efficiently via the standard 'thin' QR decomposition. If $\Phi = QR$, where Q is an orthonormal $n \times k$ matrix, then $\Pi_\phi = QQ^\top$. Therefore:

$$x^{(t+1)} = \left[QQ^\top x^{(t)} - \alpha(QQ^\top M x^{(t)} + QQ^\top q) \right]_+ \tag{6}$$

$Q^\top M$ (and $QQ^\top q$) can be precomputed, and so each iteration only takes $\mathcal{O}(kn)$ operations rather than $\mathcal{O}(n^2)$—a big reduction if $k \ll m$.

The Galerkin approximated LCP is an exact fixed point iteration for a different LCP (N, r) where $N = I - \Pi_\Phi + \alpha \Pi_\Phi M = \alpha \Pi_\Phi M + \Pi_0$ (Π_0 denotes projection onto the nullspace of Φ) and $r = \alpha \Pi_\Phi q$. This is a special kind of additive decomposition; N is a *projective* matrix and the LCP (N, r) is strictly monotone iff (M, q) was (Lemma 5.1 in [6]).

Projective monotone LCPs—a more general class than strictly monotone projective LCPs—can be solved efficiently with a variant of the unified interior point (UIP) method [7] that requires only $\mathcal{O}(nk^2)$ operations per UIP iteration [5]. UIP requires only a polynomial number of iteration to get within ϵ of the solution. Without knowledge of this projective structure then each iteration of UIP might be much more expensive.

2.2 Perturbations of Galerkin LCPs

The projection algorithm also helps us analyze how perturbing F affect the solution of strictly monotone LCPs. This occurs whenever F is estimated from data, such as when the LCP is the KKT system for either an MDP planning problem or SVM training problem.

Lem. 4. *Suppose a strictly monotone LCP (M, q) with solution x^* has been perturbed to form another strictly monotone LCP $(M + \Xi, q + \xi) = (\tilde{M}, \tilde{q})$. Let $\tilde{x}$ be the Galerkin fixed point for the perturbed LCP with respect to basis Φ. Then for any* monotonic norm $\|\cdot\|$

$$\|\tilde{x} - x^*\| \leq \frac{\alpha^2 \|\Pi_\Phi \Xi\| \|\Pi_\Phi q\|}{(1-\gamma)^2} + \frac{\alpha \|\Pi_\Phi \xi\|}{1-\gamma} + \frac{\|x^* - \Pi_\Phi x^*\|}{1-\gamma}. \tag{7}$$

Recall that α is the step size of the update operator and γ is its contraction factor. A monotonic norm $\|\cdot\|$ is one where if $|x| \leq |y|$ on a component-by-component basis, then $\|x\| \leq \|y\|$. This lemma follows from the fact that update (3) is a contraction, so both the representation error (projection onto $\mathsf{span}(\Phi)$) and perturbation error (Ξ and ξ) are contracted by γ after every iteration.

Proof. Let $\tilde{T}(x) = [\Pi_\Phi(x - \alpha(\tilde{M}x + \tilde{q}))]_+$ be the perturbed Galerkin operator with fixed point $\tilde{x}$ and let $\bar{T}(x) = [\Pi_\Phi(x - \alpha(Mx + q))]_+$ be the unperturbed Galerkin operator with fixed point $\bar{x}$. Consider the distance between the unperturbed Galerkin fixed point $\bar{x}$ and a point $x^{(t+1)}$ on perturbed update path $x^{(0)} = 0$, $x^{(i)} = \tilde{T}x^{(i-1)}$:

$$\begin{aligned}
\|x^{(t+1)} - \bar{x}\| &= \|\tilde{T}x^{(t)} - \bar{T}\bar{x}\| && (8)\\
&= \left\| \left[\Pi_\Phi(x^{(t)} - \alpha(Mx^{(t)} + q)) - \Pi_\Phi \alpha(\Xi x^{(t)} + \xi) \right]_+ - \bar{T}\bar{x} \right\| && (9)\\
&\leq \|\bar{T}x^{(t-1)} - \bar{T}\bar{x}\| + \alpha \|\Pi_\Phi \Xi x^{(t)} + \Pi_\Phi \xi\| && (10)\\
&\leq \gamma \|x^{(t-1)} - \bar{x}\| + \alpha \|\Pi_\Phi \Xi x^{(t)} + \Pi_\Phi \xi\| && (11)
\end{aligned}$$

The transition between (9) and its sequel follows from $[\cdot]_+$ being convex and 1-Lipschitz. Recursively applying (11) and allowing $t \to \infty$:

$$\|\tilde{x} - \bar{x}\| \leq (\alpha(\|\Pi_\Phi \Xi\| \|\bar{x}\| + \|\Pi_\Phi \xi\|))/(1-\gamma). \tag{12}$$

Using this inequality with $\Xi = 0, \xi = -q$ (hence $\tilde{x} = 0$) yields $\|\bar{x}\| \leq \alpha \|\Pi_\Phi q\|/(1-\gamma)$. Combining (12) and the bound on $\|\bar{x} - x^*\|$ from Lemma 3 with the triangle inequality yields (7). □

Lemma 4 only holds when both the original LCP and the perturbed one are strictly monotone. This is a reasonable assumption in settings where a class of problems is strictly monotone, and the class is closed under some perturbation model. For example, if a class of regularized MDP planning problems has strictly monotone LCPs, then any perturbation to an instance's costs or transition matrix could be analyzed with the above result.

Lemma 4 is appealing because we bound using only the projection of the measurement error onto $\mathsf{span}(\Phi)$, which allows us to ignore any error orthogonal to Φ. The Galerkin approximation may have nice statistical properties when x^* is well approximated by a low rank basis, but F perturbed by full dimensional noise.

Lemma 4 can be used with $\Pi_\Phi = I$ to get perturbation results for arbitrary strictly monotone LCP. These bounds can be compared to the monotone LCP error bounds found in [8]. Our results currently only apply to strictly monotone LCPs, but our bound has the virtue of being concrete and computable, rather than the non-constructive bounds provided in said paper.

3 Conclusions

In this paper we presented a number of results about a Galerkin method for strictly monotone LCPs. We first presented a basic projective fixed point algorithm, before modifying it to work for a Galerkin approximation of an LCP. Not only can this Galerkin approximation be solved more efficiently than the original system, but this paper proves that the Galerkin approximation may also have nice statistical properties—the approximation is more robust to perturbations than the original system.

Future directions of research include expanding the class of LCPs that we are able to solve and analyze. Broader classes of LCPs that are still polynomial include monotone LCPs (*i.e.* positive semidefinite M) and a generalization of monotone LCPs based on a type of matrix called $P_*(\kappa)$ [7]. We are also working on stochastic Galerkin methods to reduce the cost of each iteration from $\mathcal{O}(kn)$ to $\mathcal{O}(km)$ for $k, m \ll n$.

References

[1] Dimitri P Bertsekas. Projected equations, variational inequalities, and temporal difference methods. *Lab. for Information and Decision Systems Report LIDS-P-2808, MIT*, 2009.

[2] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[3] R. Cottle, J.S. Pang, and R.E. Stone. *The linear complementarity problem*. Classics in applied mathematics. SIAM, 1992.

[4] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 1. Springer, 2003.

[5] Geoffrey J. Gordon. Fast solutions to projective monotone linear complementarity problems. arXiv:1212.6958, 2012.

[6] Geoffrey J. Gordon. Galerkin methods for complementarity problems and variational inequalities. arXiv:1306.4753, 2013.

[7] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A unified approach to interior point algorithms for linear complementarity problems*. Lecture Notes in Computer Science. Springer, 1991.

[8] O.L. Mangasarian and J. Ren. New improved error bounds for the linear complementarity problem. *Mathematical Programming*, 66(1-3):241–255, 1994.