Semidefinite projections, regularization algorithms, and polynomial optimization

Jérôme MALICK

CNRS, Grenoble, France

SIAM Conf on Optimization – Darmstadt, May 2011

Semidefinite projections: examples, algorithms, applications

- This talk sketches the content of Henrion-Malick '11

D. Henrion and J. Malick
Chapter *"Projection methods in conic optimization"*
Handbook of conic programming and polynomial optimization, 2011
Editors: M. Anjos and J.B Lasserre

- Projections onto subsets of $\mathcal{S}_n^+$: examples, algorithms, applications

- Review of material of papers; among those:

  Malick '04, Qi-Sun '06, Malick-Povh-Rendl-Wiegele '07,
  Zhao-Sun-Toh '08, Henrion-Malick '09, Nie '09, ...

- Numerical experiments are just illustrations (in Matlab)
  (no extensive comparison, no benchmarking,... refer to above papers)

- (Pedagogical) presentation: pointing out, clarifying, unifying ideas...
  showing common techniques (semidefinite projections!)

Outline

1. Semidefinite projections, algorithms
2. Illustration: SDP and SOS feasibility
3. Projections in regularization methods for SDP
4. Illustration: polynomial optimization

Outline

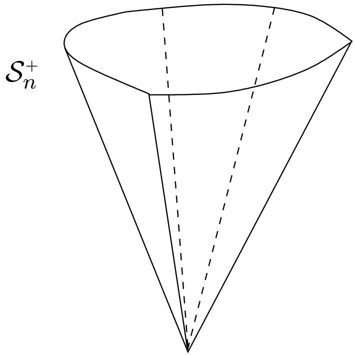
1 **Semidefinite projections, algorithms**

2 Illustration: SDP and SOS feasibility

3 Projections in regularization methods for SDP

4 Illustration: polynomial optimization

Cone of positive semidefinite matrices



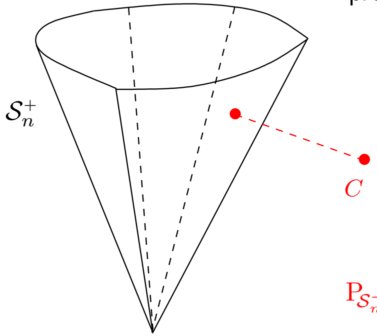
$\mathcal{S}_n$	:	the space of symmetric matrices
$\langle \cdot, \cdot \rangle$	:	usual inner product (Frobenius)
$\| \cdot \|$	:	associated norm

The cone of positive semidefinite matrices

$$\begin{aligned} \mathcal{S}_n^+ &= \{A \in \mathcal{S}_n, \ \ \forall x \in \mathbb{R}^n, \ x^\top A x \geqslant 0\} \\ &= \{A \in \mathcal{S}_n, \ \ \lambda_{\min}(A) \geqslant 0\} \end{aligned}$$

$\mathcal{S}_n^+$ is closed and convex, with nice properties, see eg Wolkowicz *et al* '00

Projection onto the (closed convex) cone



Well-known result: explicit expression of the projection onto the cone (eg Higham '88)

$$C = P \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} P^\top$$

is projected onto

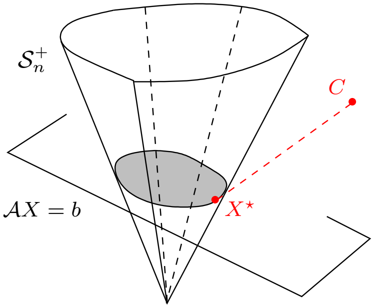
$$\mathrm{P}_{\mathcal{S}_n^+}(C) = P \begin{bmatrix} \max\{0, \lambda_1\} & & \\ & \ddots & \\ & & \max\{0, \lambda_n\} \end{bmatrix} P^\top$$

$\longrightarrow$ computational cost: eigendecomposition

Projection onto subsets of the cone

subsets $= \mathcal{S}_n^+ \cap \{X : \mathcal{A}X = b\}$

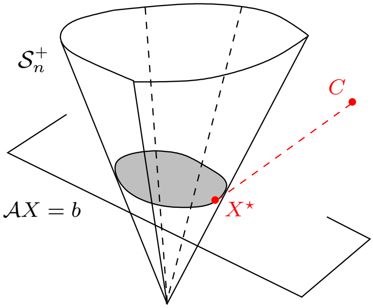
with $b \in \mathbb{R}^m$ and $\mathcal{A}\colon \mathcal{S}_n \to \mathbb{R}^m$



Projection onto subsets of the cone

subsets $= \mathcal{S}_n^+ \cap \{X : \mathcal{A}X = b\}$

with $b \in \mathbb{R}^m$ and $\mathcal{A}\colon \mathcal{S}_n \to \mathbb{R}^m$



Pb: "Semidefinite least-squares"

$$\begin{cases} \min \quad \|X - C\|^2 \\ \quad \mathcal{A}X = b \\ \quad X \succcurlyeq 0 \end{cases}$$

Example: nearest correlation matrix (Higham '02)

$$\begin{cases} \min \quad \|X - C\|^2 \\ \quad \operatorname{diag} X = e \\ \quad X \succcurlyeq 0 \end{cases}$$

Applications: Linear Algebra, Optics, Control, Statitics, Finance...
see a list in Henrion-Malick '11

Computing semidefinite projections

1. First idea: reformulate SDLS as a linear conic program

$$\begin{cases} \min \quad t \\ \quad \|X - C\| \leqslant t \\ \quad \mathcal{A}X = b, \ X \succcurlyeq 0 \end{cases}$$

with usual conic solvers (`SeDumi`, `SDPT3`,...) $\longrightarrow$ no good results

Computing semidefinite projections

1. First idea: reformulate SDLS as a linear conic program

$$\begin{cases} \min \quad t \\ \quad \|X - C\| \leqslant t \\ \quad \mathcal{A}X = b, \ X \succcurlyeq 0 \end{cases}$$

with usual conic solvers (`SeDumi`, `SDPT3`,...) $\longrightarrow$ no good results

2. Dedicated methods:

1. alternating projections + correction (Higham '02)

$$\begin{cases} X_{k+1} = \mathrm{P}_{\mathcal{S}_n^+}(Z_k), \quad Y_{k+1} = \mathrm{P}_{\{\mathcal{A}X=b\}}(X_{k+1}) \\ Z_{k+1} = Z_k - (X_{k+1} - Y_{k+1}) \end{cases}$$

2. by duality (Malick '04, Qi-Sun '06, Borsdorf-Higham '08...)
3. interior-point method (Toh-Todd-Tutuncu '06)
4. alternating directions (He-Xu-Yian '11)

$$\begin{cases} X_{k+1} = \mathrm{P}_{\mathcal{S}_n^+}\left(\frac{\beta Y_k + Z_k + C}{1+\beta}\right), \quad Y_{k+1} = \mathrm{P}_{\{\mathcal{A}X=b\}}\left(\frac{\beta X_{k+1} - Z_k + C}{1+\beta}\right) \\ Z_{k+1} = Z_k - \beta(X_{k+1} - Y_{k+1}) \end{cases}$$

Dual approach

1. Apply standard machinery of Lagrangian duality

$$\theta(\lambda) = \begin{cases} \min & \|X - C\|^2 - y^\top(\mathcal{A}X - b) \\ & X \succcurlyeq 0 \end{cases}$$

$$\begin{cases} \max & \theta(y) \\ & \lambda \in \mathbb{R}^m \end{cases}$$

dual problem is concave
and differentiable !

Dual approach

1. Apply standard machinery of Lagrangian duality

$$\theta(\lambda) = \begin{cases} \min & \|X - C\|^2 - y^\top(\mathcal{A}X - b) \\ & X \succcurlyeq 0 \end{cases}$$

$$\begin{cases} \max & \theta(y) \\ & \lambda \in \mathbb{R}^m \end{cases}$$

dual problem is concave and differentiable !

2. Apply standard algorithms: $y_{k+1} = y_k - \tau_k W_k \nabla\theta(y_k)$

- Steepest descent = alternating projections (Henrion-Malick '09)
- Quasi-Newton (Malick '04)
- Generalized Newton (Qi-Sun '06)
  $\longrightarrow$ quadratic convergence (under non-degeneracy assumption)

Dual approach

1. Apply standard machinery of Lagrangian duality

$$\theta(\lambda) = \begin{cases} \min & \|X - C\|^2 - y^\top(\mathcal{A}X - b) \\ & X \succcurlyeq 0 \end{cases}$$

$$\begin{cases} \max & \theta(y) \\ & \lambda \in \mathbb{R}^m \end{cases}$$

dual problem is concave and differentiable !

2. Apply standard algorithms: $y_{k+1} = y_k - \tau_k W_k \nabla\theta(y_k)$

- Steepest descent = alternating projections (Henrion-Malick '09)
- Quasi-Newton (Malick '04)
- Generalized Newton (Qi-Sun '06)
  $\longrightarrow$ quadratic convergence (under non-degeneracy assumption)

4. Stopping criteria: $0 \approx \|\nabla\theta(\lambda_k)\| = \|\mathcal{A}X_k - b\|$ (primal infeasibility)

3. Under primal Slater: no duality gap and we get the projection

$\longrightarrow$ (Robust, fast) algorithms... (Eg. nearest correlation: $n = 5000$, 1h)

Generalisations, applications

Bottomline: We can compute semidefinite projections efficiently

Semidefinite projections are inner algorithms of more involved algorithms

- Proximal methods for (linear) SDP (Malick *et al* '07) – **Part 3**

$$\begin{cases} \min & \langle C, X \rangle \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Augmented Lagrangian for weighted projections (Qi-Sun '08)

$$\begin{cases} \min & \frac{1}{2} \sum_{i,j=1}^{n} H_{ij}(X_{ij} - C_{ij})^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Penalty/Augmented Lagrangian for low-rank (Li-Qi '11, Gao-Sun '11)

$$\begin{cases} \min & \frac{1}{2} \|X - C\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0, \ \operatorname{rank} X = r \end{cases}$$

Outline

1. Semidefinite projections, algorithms
2. **Illustration: SDP and SOS feasibility**
3. Projections in regularization methods for SDP
4. Illustration: polynomial optimization

Semidefinite feasibility problem

- Basic SDP feasibility problem:

$$\begin{cases} \mathcal{A}X = b \\ X \succcurlyeq 0 \end{cases}$$

- Usually solved by linear semidefinite optimization solvers

$$\begin{cases} \min \quad \langle C, X \rangle \\ \quad \mathcal{A}X = b \\ \quad X \succcurlyeq 0 \end{cases}$$

e.g. with `SeDuMi` (interior-point) with $C = 0$

Semidefinite feasibility problem

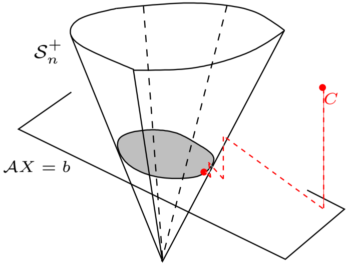
- Basic SDP feasibility problem:

$$\begin{cases} \mathcal{A}X = b \\ X \succcurlyeq 0 \end{cases}$$

- Usually solved by linear semidefinite optimization solvers

$$\begin{cases} \min \quad \langle C, X \rangle \\ \quad \mathcal{A}X = b \\ \quad X \succcurlyeq 0 \end{cases}$$

e.g. with SeDuMi (interior-point) with $C = 0$



- General convex feasibility problems solved by (improved) alternating projections

Semidefinite feasibility problem

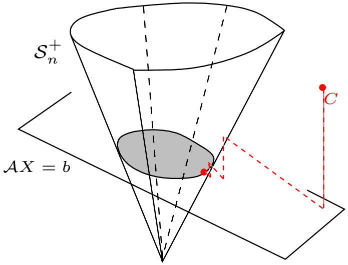
- Basic SDP feasibility problem:

$$\begin{cases} \mathcal{A}X = b \\ X \succcurlyeq 0 \end{cases}$$

- Usually solved by linear semidefinite optimization solvers

$$\begin{cases} \min \quad \langle C, X\rangle \\ \quad \mathcal{A}X = b \\ \quad X \succcurlyeq 0 \end{cases}$$

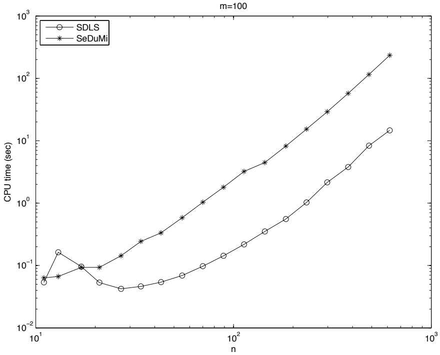
e.g. with `SeDuMi` (interior-point) with $C = 0$



- General convex feasibility problems solved by (improved) alternating projections
- (Trivial) idea: why not projecting C ? to be computed by dual algorithms ? (Henrion Malick '09)
- Note: Slater has natural appeal as (metric) regularity...

Illustration: random SDP feasibility

- Random (dense) SDP feasibility (with Slater point)
- $n = 10, \ldots, 1000$ (size of X) and $m = 100$ fixed (# of constraints)
- Without particular knowledge on the problem: project $C = 0$...
- Medium accuracy: $\varepsilon = 10^{-6}$
- Comparison CPU time: – `SeDuMi` (linear SDP)
  – 50 line Matlab dual methods (projection)



Polynomials and sum-of-squares

- Polynomial of degree $2d$: $p(v) = \sum_{\alpha_1+\cdots+\alpha_N \leqslant 2d} p_\alpha \, v_1^{\alpha_1} \cdots v_N^{\alpha_N}$
- Polynomial p is a sum-of-square (SOS) if: $p(v) = \sum_{i=1}^{r} q_i(v)^2$
- Let $\pi(v)$ be vector of basis of polynomials of degree $\leqslant d$

$$p \text{ SOS} \iff p(v) = \langle X, \pi(v)\pi(v)^\top \rangle \text{ with } X \succcurlyeq 0$$

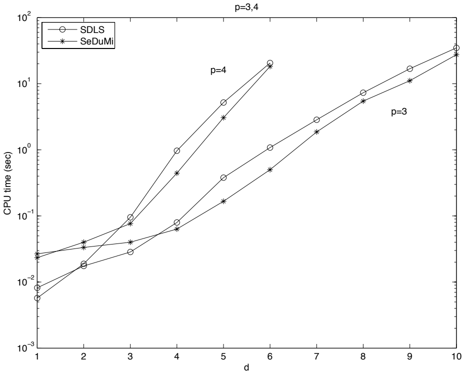
- Example: $N = 1$, $d = 2$, $\pi(v) = [1, v, v^2]^\top$

$$v^4 + 2\,v^2 + 1 \;=\; \pi(v)^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \pi(v) \;=\; \pi(v)^\top \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \pi(v)$$

- Testing if p is a SOS = SDP feasibility $\boxed{\mathcal{A}X = b \text{ with } X \succcurlyeq 0}$
  ($\mathcal{A}$ depends on π, b on p)
- n (size of X) and m (# of constraints) explode with N and d

Illustration: SOS feasibility problems

- Using GloptiPoly (Henrion-Lasserre)
- Generate random SOS polynomial (with Slater point)
- $N = 3, 4$ (number of variables) and $d = 1, \ldots, 10$ (degree of p)
- `SeDuMi` and 50 line Matlab code comparable



- **Question:** which C to project ? better than $C = 0$... see **Part 4**

Outline

1 Semidefinite projections, algorithms

2 Illustration: SDP and SOS feasibility

3 **Projections in regularization methods for SDP**

4 Illustration: polynomial optimization

Semidefinite programming (SDP)

- Standard linear semidefinite programming

$$(\text{SDP}) \begin{cases} \min & \langle C, X\rangle \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Many efficient solvers based on different approaches
  1. primal-dual interior point methods (eg Todd '01 for a review)
  2. modified barrier method (PENSDP Kocvara-Stingl '07)
  3. spectral bundle methods, $\lambda_{\max}(X)$ (Helmberg-Rendl '00)
  4. low-rank methods, $X = RR^\top$ with $R \in \mathbb{R}^{n\times r}$ (Burer-Monteiro '03)
  5. and others ! Sorry for not citing all of them...

Semidefinite programming (SDP)

- Standard linear semidefinite programming

$$(\mathrm{SDP}) \begin{cases} \min & \langle C, X\rangle \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Many efficient solvers based on different approaches
  1. primal-dual interior point methods (eg Todd '01 for a review)
  2. modified barrier method (PENSDP Kocvara-Stingl '07)
  3. spectral bundle methods, $\lambda_{\max}(X)$ (Helmberg-Rendl '00)
  4. low-rank methods, $X = RR^\top$ with $R \in \mathbb{R}^{n\times r}$ (Burer-Monteiro '03)
  5. and others ! Sorry for not citing all of them...

- Relaxations of combinatorial optimization or polynomial optimization problems are challenging problems...

- Malick-Povh-Rendl-Wiegele '07 introduce an approach using semidefinite projections (regularization algorithms)

- Target problems: n not too big, m very large:

$$n \leqslant 1000 \qquad m \text{ of order } 100,000$$

Semidefinite projections in primal proximal method

- Consider the SDP problem

$$\begin{cases} \min \quad \langle C, X \rangle \\ \quad \mathcal{A}X = b \\ \quad X \succcurlyeq 0 \end{cases}$$

Semidefinite projections in primal proximal method

- Consider the SDP problem with a regularization term

$$\left\{\begin{array}{ll} \min & \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{array}\right.$$

Semidefinite projections in primal proximal method

- Consider the SDP problem with a regularization term

$$\mathrm{Prox}(Y) := \begin{cases} \min & \langle C, X \rangle + \frac{1}{2t}\|X - Y\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

Semidefinite projections in primal proximal method

- Consider the SDP problem with a regularization term

$$\mathrm{Prox}(Y) := \begin{cases} \min & \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Classical technique of convex analysis: proximal method (Bellman '66, Martinet '70, Rockafellar '76, and many others after)

- Y solution of linear SDP $\iff Y = \mathrm{Prox}(Y)$

- Fixed-point algorithm: $Y_{k+1} = \mathrm{Prox}(Y_k)$ (in fact $Y_{k+1} \approx \mathrm{Prox}(Y_k)$)

- Dual interpretation: augmented Lagrangian
  Related to augmented Lagrangian: BPM (Rendl *et al* '07), primal (Burer-Vandebusshe '06), primal-dual (Jarre-Rendl '07)

Semidefinite projections in primal proximal method

- Consider the SDP problem with a regularization term

$$\mathrm{Prox}(Y) := \begin{cases} \min & \langle C, X\rangle + \frac{1}{2t}\|X - Y\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{cases}$$

- Classical technique of convex analysis: proximal method (Bellman '66, Martinet '70, Rockafellar '76, and many others after)

- Y solution of linear SDP $\iff$ $Y = \mathrm{Prox}(Y)$

- Fixed-point algorithm: $Y_{k+1} = \mathrm{Prox}(Y_k)$ (in fact $Y_{k+1} \approx \mathrm{Prox}(Y_k)$)

- Dual interpretation: augmented Lagrangian
  Related to augmented Lagrangian: BPM (Rendl *et al* '07), primal (Burer-Vandebusshe '06), primal-dual (Jarre-Rendl '07)

- Malick-Povh-Rendl-Wiegele '07 introduces a family of regularization algorithms for SDP depending on
  - the inner algorithm to compute Prox (= semidefinite projection)
  - a rule to stop this inner algorithm
  - a rule to update the prox-parameter t

Regularization algorithms

Algorithm

Outer loop on k until $\|Y_{k+1} - Y_k\|$ small:
  Inner loop on ℓ until $\|b - \mathcal{A}X_\ell\|$ small enough:
    Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ (and Z_ℓ) and $g_\ell = b - \mathcal{A}X_\ell$
    Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ with appropriate τ_ℓ and W_ℓ
  end (inner loop)
  Update $Y_{k+1} = X_\ell$
end (outer loop)

Regularization algorithms

Algorithm

Outer loop on k *until* $\|Y_{k+1} - Y_k\|$ *small:*
  Inner loop on ℓ *until* $\|b - \mathcal{A}X_\ell\|$ *small enough:*
    Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ *(and* Z_ℓ*) and* $g_\ell = b - \mathcal{A}X_\ell$
    Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ *with appropriate* τ_ℓ *and* W_ℓ
  end (inner loop)
  Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)

Regularization algorithms

Algorithm

Outer loop on k until $\|Y_{k+1} - Y_k\|$ small:
Inner loop on ℓ until $\|b - \mathcal{A}X_\ell\|$ small enough:
Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ (and Z_ℓ) and $g_\ell = b - \mathcal{A}X_\ell$
Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ with appropriate τ_ℓ and W_ℓ
end (inner loop)
Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)
- Inner stopping test and outer stopping test

Regularization algorithms

Algorithm

Outer loop on k *until* $\|Z + \mathcal{A}^* y - C\|$ *small:*
Inner loop on ℓ *until* $\|b - \mathcal{A}X_\ell\|$ *small enough:*
Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ *(and* Z_ℓ*) and* $g_\ell = b - \mathcal{A}X_\ell$
Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ *with appropriate* τ_ℓ *and* W_ℓ
end (inner loop)
Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)
- Inner stopping test and outer stopping test ($=$ dual infeasibility)

Regularization algorithms

Algorithm

Outer loop on k *until* $\|Z + \mathcal{A}^* y - C\|$ *small:*
Inner loop on ℓ *until* $\|b - \mathcal{A}X_\ell\|$ *small enough:*
Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ *(and* Z_ℓ*) and* $g_\ell = b - \mathcal{A}X_\ell$
Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ *with appropriate* τ_ℓ *and* W_ℓ
end (inner loop)
Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)
- Inner stopping test and outer stopping test ($=$ dual infeasibility)
- Regularization algorithms (projections methods) $\perp$ interior points
  Complementary and semidefiniteness are ensured throughout

Regularization algorithms

Algorithm

Outer loop on k until $\|Z + \mathcal{A}^* y - C\|$ small:
Inner loop on ℓ until $\|b - \mathcal{A}X_\ell\|$ small enough:
Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ (and Z_ℓ) and $g_\ell = b - \mathcal{A}X_\ell$
Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ with appropriate τ_ℓ and W_ℓ
end (inner loop)
Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)
- Inner stopping test and outer stopping test ($=$ dual infeasibility)
- Regularization algorithms (projections methods) $\perp$ interior points
  Complementary and semidefiniteness are ensured throughout
- Under some technical assumptions:

  any accumulation point of the generated sequence is solution

Regularization algorithms

Algorithm

Outer loop on k *until* $\|Z + \mathcal{A}^* y - C\|$ *small:*
Inner loop on ℓ *until* $\|b - \mathcal{A}X_\ell\|$ *small enough:*
Compute $X_\ell = \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A}^* y_\ell - C))$ *(and* Z_ℓ*) and* $g_\ell = b - \mathcal{A}X_\ell$
Update $y_{\ell+1} = y_\ell + \tau_\ell W_\ell g_\ell$ *with appropriate* τ_ℓ *and* W_ℓ
end (inner loop)
Update $Y_{k+1} = X_\ell$
end (outer loop)

- Low memory: $\mathcal{A}$, $\mathcal{A}^*$ and use low-memory methods for dual projection subproblems (QN, N-CG)
- Inner stopping test and outer stopping test ($=$ dual infeasibility)
- Regularization algorithms (projections methods) $\perp$ interior points
  Complementary and semidefiniteness are ensured throughout
- Under some technical assumptions:

  any accumulation point of the generated sequence is solution
- Question: when to stop inner iterations ?

Stopping inner computations (semidefinite projections)

1. Solving inner problem to (approx.) optimality (Zhao-Sun-Toh '08)
   $\longrightarrow$ proofs of inner and outer convergence + good performance

Stopping inner computations (semidefinite projections)

1. Solving inner problem to (approx.) optimality (Zhao-Sun-Toh '08)
   $\longrightarrow$ proofs of inner and outer convergence + good performance

2. Only one iteration (Malick-Povh-Rendl-Wiegele '07)
   - Linear SDP is equivalent to min-max

$$\left\{ \begin{array}{l} \min \\ \quad Y \in \mathcal{S}_n \end{array} \left(\begin{array}{l} \max \quad b^\top y - \frac{1}{2t}(\|Y\|^2 - \|\operatorname{P}_{\mathcal{S}_n^+}(Y + t(\mathcal{A}^* y - C))\|^2 \\ \quad y \in \mathbb{R}^m \end{array} \right) \right.$$

Stopping inner computations (semidefinite projections)

1. Solving inner problem to (approx.) optimality (Zhao-Sun-Toh '08)
   $\longrightarrow$ proofs of inner and outer convergence + good performance

2. Only one iteration (Malick-Povh-Rendl-Wiegele '07)
   - Linear SDP is equivalent to min-max

$$\left\{ \begin{array}{l} \min \\ \quad Y \in \mathcal{S}_n \end{array} \left(\begin{array}{ll} \max & b^\top y - \frac{1}{2t}(\|Y\|^2 - \|\operatorname{P}_{\mathcal{S}_n^+}(Y + t(\mathcal{A}^* y - C))\|^2 \\ \quad y \in \mathbb{R}^m & \end{array} \right) \right.$$

   - Keeping $W_k = [\mathcal{A}\mathcal{A}^*]$ (constant) and $\tau_k = 1/t_k$ gives a simple regularization method with only one loop

$$\begin{array}{rcl} Y_{k+1} & = & \operatorname{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A} y_k - C)) \\ y_{k+1} & = & y_k + [\mathcal{A}\mathcal{A}^*]^{-1}(b - \mathcal{A} Y_k)/t_k. \end{array}$$

   - Note: $\mathcal{A}\mathcal{A}^*$ (and Cholesky factorization) computed once
   - Corresponds to basic block-coordinate method Wen-Goldfarb-Yin '10

Stopping inner computations (semidefinite projections)

1. Solving inner problem to (approx.) optimality (Zhao-Sun-Toh '08)
   $\longrightarrow$ proofs of inner and outer convergence + good performance

2. Only one iteration (Malick-Povh-Rendl-Wiegele '07)
   - Linear SDP is equivalent to min-max

$$\left\{ \begin{array}{l} \min \\ \quad Y \in \mathcal{S}_n \end{array} \left(\begin{array}{l} \max \quad b^\top y - \frac{1}{2t}(\|Y\|^2 - \| \mathrm{P}_{\mathcal{S}_n^+}(Y + t(\mathcal{A}^* y - C))\|^2 \\ \quad y \in \mathbb{R}^m \end{array} \right) \right.$$

   - Keeping $W_k = [\mathcal{A}\mathcal{A}^*]$ (constant) and $\tau_k = 1/t_k$ gives a simple regularization method with only one loop

$$\begin{array}{rcl} Y_{k+1} & = & \mathrm{P}_{\mathcal{S}_n^+}(Y_k + t_k(\mathcal{A} y_k - C)) \\ y_{k+1} & = & y_k + [\mathcal{A}\mathcal{A}^*]^{-1}(b - \mathcal{A} Y_k)/t_k. \end{array}$$

   - Note: $\mathcal{A}\mathcal{A}^*$ (and Cholesky factorization) computed once
   - Corresponds to basic block-coordinate method Wen-Goldfarb-Yin '10

3. Something in-between ??
   - a first study by Fuentes-Malick-Lemaréchal '10...
   - Note: Zhao-Sun-Toh '08 uses 2nd strategy as pre-processing

Outline

1. Semidefinite projections, algorithms
2. Illustration: SDP and SOS feasibility
3. Projections in regularization methods for SDP
4. **Illustration: polynomial optimization**

Regularization methods for polynomial optimization

- Henrion Malick '11: illustration by comparison between three codes
  - advanced "regularization" SDPNAL (optimistic) (Mex-files,...)
  - simple regularization mprw (cautious) (50 lines of Matlab)
  - interior-point SeDuMi

  on relaxations of polynomial problems; for more, see Nie '09

Regularization methods for polynomial optimization

- Henrion Malick '11: illustration by comparison between three codes
  - advanced "regularization" SDPNAL (optimistic) (Mex-files,...)
  - simple regularization mprw (cautious) (50 lines of Matlab)
  - interior-point SeDuMi

  on relaxations of polynomial problems; for more, see Nie '09

- **Testing positivity of polynomials**
  - Difficult problem: testing if $p(v) \geqslant 0$ for all $v \in \mathbb{R}^N$
  - Relaxation: testing p SOS $\iff$ SDP feasibility ($\mathcal{A}X = b$ with $X \succcurlyeq 0$)

- **Unconstrained minimization of a polynomial**
  - Difficult problem

$$\begin{cases} \min & p(v) \\ & v \in \mathbb{R}^N \end{cases} \qquad \iff \qquad \begin{cases} \max & \gamma \\ & p(v) - \gamma \geqslant 0 \text{ for all } v \in \mathbb{R}^N \end{cases}$$

  - Relaxation: Tractable problem

$$\begin{cases} \max & \gamma \\ & p(v) - \gamma \ \text{ SOS} \end{cases} \qquad \iff \qquad \begin{cases} \min & \langle C, X \rangle \\ & \mathcal{A}X = b, \ X \succcurlyeq 0 \end{cases}$$

- Sizes n and m explode with d and N... but $\mathcal{A}\mathcal{A}^*$ diagonal !

Illustration random SOS feasibility

- Degree 6 full-rank polynomials (Slater)

N	n	m	SeDuMi	mprw	SDPNAL
8	165	3003	25	0.35	0.16
9	220	5005	110	0.66	0.25
10	286	8008	410	1.3	0.43
11	364	12376	1500	3.0	0.73
12	455	18564	> 3600	5.0	1.3

- Low-rank polynomial (no Slater)

N	n	m	SeDuMi	mprw	SDPNAL
8	165	3003	61	4.8	0.98
9	220	5005	330	12	1.2
10	286	8008	1300	24	2.5
11	364	12376	> 3600	50	3.5
12	455	18564	> 3600	110	6.6

- Times in seconds (with 2 digits); tolerance 10^{-9}...
- Again: just illustration, no benchmarking !

Unconstrained polynomial optimization

- Random well-behaved instance: $p(v) = p_0(v) + \sum_{i=1}^{N} v_i^{2d}$

N	n	m	SeDuMi	MPRW	SDPNAL
5	21	126	0.09	0.05	0.18
10	66	1000	1.9	0.45	0.29
15	136	3875	74	3.0	0.68

- Structured example (for more, see Nie '09)

$$p(v) := \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{i}(v_j + {v_j}^2)\right)^2 + \left(1 - \sum_{j=i}^{N}(v_j + {v_j}^3)\right)^2$$

N	n	m	SeDuMi	MPRW	SDPNAL
10	286	8007	1800	200	71
11	364	12375	7162	490	150
12	455	18563	> 7200	1500	530
13	560	27131	> 7200	3500	2300
14	680	38760	> 7200	> 7200	9900

Last slide

- Semidefinite optimization and projections

$$\left\{ \begin{array}{ll} \min & \langle C, X \rangle \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{array} \right. \qquad \left\{ \begin{array}{ll} \min & \|X - C\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{array} \right.$$

- Dual algorithms for computing projections

$\longrightarrow$ use nice dual properties
$\longrightarrow$ are efficient, cope with large-scale

- Regularization algorithms for SDP (primal proximal, dual augmented)

$\longrightarrow$ opposite to Interior Points
$\longrightarrow$ promising approach for polynomial optimization
$\longrightarrow$ much more (theoretical, algorithmic, implementation) work to do

- (Smooth) introduction + references

D. Henrion and J. Malick
Chapter *"Projection methods in conic optimization"*
Handbook of conic programming and polynomial optimization, 2011
Editors: M. Anjos and J.B Lasserre

Last slide

- Semidefinite optimization and projections

$$\left\{\begin{array}{ll} \min & \langle C, X\rangle \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{array}\right. \qquad \left\{\begin{array}{ll} \min & \|X - C\|^2 \\ & \mathcal{A}X = b \\ & X \succcurlyeq 0 \end{array}\right.$$

- Dual algorithms for computing projections

$\longrightarrow$ use nice dual properties

$\longrightarrow$ are efficient, cope with large-scale

- Regularization algorithms for SDP (primal proximal, dual augmented)

$\longrightarrow$ opposite to Interior Points

$\longrightarrow$ promising approach for polynomial optimization

$\longrightarrow$ much more (theoretical, algorithmic, implementation) work to do

- (Smooth) introduction + references

D. Henrion and J. Malick
Chapter *"Projection methods in conic optimization"*
Handbook of conic programming and polynomial optimization, 2011
Editors: M. Anjos and J.B Lasserre

thanks !