Symmetric Cone Programming with Applications to Finance.

A. d'Aspremont*

Stanford University

Mathematical Finance, AMAM 2003.

Saturday, February 15, 2003.

*Email: alexandre.daspremont@polytechnique.org

1 Introduction

- Convexity $\Longrightarrow$ low complexity:

  "... *In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.*" T. Rockafellar.

- True: Nemirovskii & Yudin (1979).

- Very true: Karmarkar (1984).

- Seriously true: structured convex programming, Nesterov & Nemirovskii (1994).

1.1 Standard convex complexity analysis

- All convex minimization problems with a first order oracle (returning $f(x)$ and a subgradient) can be solved in polynomial time in size and number of precision digits.

- Proved using the *ellipsoid method* by Nemirovskii & Yudin (1979).

- Very slow convergence in practice.

1.2 Linear Programming

- Simplex algorithm by Dantzig (1949): exponential worst-case complexity, very efficient in most cases.

- Khachiyan (1979) then used the ellispoid method to show the polynomial complexity of LP.

- Karmarkar (1984) describes the first efficient polynomial time algorithm for LP, using interior point methods.

1.3 From LP to structured convex programs

- Nesterov & Nemirovskii (1994) show that the interior point methods used for LPs can be applied to a larger class of structured convex problems.

- The *self-concordance* analysis that they introduce extends the polynomial time complexity proof for LPs.

- Most operations that preserve convexity also preserve self-concordance.

- The complexity of a certain number of elementary problems can be directly extended to a much wider class.

1.4 Symmetric cone programs

- An important particular case: linear programming on symmetric cones

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax - b \in \mathcal{K} \end{array}$$

- These include the LP, second-order (Lorentz) and semidefinite cone:

$$\begin{array}{ll} \text{LP:} & \{x \in \mathbb{R}^n : x \geq 0\} \\ \text{Second order:} & \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : \|x\| \leq y\} \\ \text{Semidefinite:} & \{X \in \mathsf{S}^n : X \succeq 0\} \end{array}$$

- Again, the class of problems that can be represented using these cones is extremely vast.

1.5 Sources

- Nesterov & Nemirovskii (1994), "Interior Point Polynomial Algorithms in Convex Programming", SIAM.

- Ben-Tal & Nemirovski (2001), "Lectures On Modern Convex Optimization: Analysis, Algorithms, And Engineering Applications", SIAM.

  `http://iew3.technion.ac.il/Labs/Opt/index.php?4`

- Boyd & Vandenberghe (2003), "Convex optimization", to appear.

  `http://www.stanford.edu/~boyd/cvxbook.html`

1.6 Outline

- Self-concordance and the complexity of Newton's method.
- Symmetric Cone Programming, examples & applications.
- Harder problems...

2 Newton's method for self-concordant functions

We want to solve the following unconstrained program:

$$\begin{array}{ll}\text{minimize} & f(x)\\ & x \in \mathbb{R}^n\end{array}$$

where for simplicity here f is convex with $f \in C^2(\mathbb{R}^n)$ and $\nabla^2 f(x) \succeq mI$, for some $m > 0$. Let x_0 be a initial point and let us note S the (bounded) sublevel set

$$S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

This means in particular that $\nabla^2 f(x)$ is Lipschitz continuous

$$\left\|\left(\nabla^2 f(x) - \nabla^2 f(y)\right) h\right\| \leq L \|h\| \|x - y\|, \quad \text{for } x, y, h \in \mathbb{R}^n,$$

and that there is some $M > 0$ such that

$$\nabla^2 f(x) \preceq MI \text{ on } S.$$

2.1 Newton's method

From the initial point x_0, the *Newton step* $\Delta x = x_{i+1} - x_i$ is computed as:

$$\Delta x = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x),$$

the Newton algorithm then converges in *two phases*:

- The *damped phase*: when $\|\nabla f(x)\| \geq \frac{2m^2}{L}$, we have:

$$f(x_{i+1}) - f(x_i) \leq -\gamma(m, L) \quad \text{for some } \gamma(m, L) > 0.$$

- The *pure Newton phase*: when $\|\nabla f(x)\| < \frac{2m^2}{L}$, the convergence is quadratic (the # of digits of accuracy doubles at each iteration).

This shows polynomial convergence in the strictly convex case.

2.2 What's wrong with this analysis?

- The Newton step can be interpreted as a steepest descent step in the geometry defined by the Hessian.

- In particular, the Newton step is *affine invariant*.

- It solves quadratic problems in one step.

In contrast to this, the complexity analysis that was made in the strictly convex case is *heavily dependent on a particular choice of geometry* (which has nothing to do with f).

- In practice, the constants m and L cannot be accessed and the method gives a poor indication of the actual convergence rate.

- The strict conversity assumption is unnecessarily restrictive.

2.3 Self-concordance

A better way of characterizing Lipschitz continuity of the Hessian...

Definition 1 *A function* f *defined on an open convex set* $\mathcal{C} \subset \mathbb{R}^n$ *is called self-concordant with parameter* a *iff*

$$\left|D^3 f(x)[h,h,h]\right| \leq 2a^{-\frac{1}{2}} \left(h^T \nabla^2 f(x) h\right)^{\frac{3}{2}}$$

A function is then called *strongly self-concordant* iff its sublevel sets

$$\{x \in \mathcal{C} : f(x) \leq t\}$$

are closed for all $t \in \mathbb{R}$. This implies in particular that $f(x) \rightarrow \infty$ on the boundary of $\mathcal{C}$.

2.4 Newton's method on s.c. functions

In this case, the two phases in Newton's method become:

- The *damped phase*: when $\lambda(f,x) = \|\nabla f(x)\|_{\nabla^2 f(x)} \geq \lambda_*$, we have:
$$f(x_{i+1}) - f(x_i) \leq a\,(\lambda_* - \ln(1+\lambda_*))$$
- The *pure Newton phase*: when $\|\nabla f(x)\| < \lambda_*$, the convergence is quadratic (the # of digits of accuracy doubles at each iteration).

With
$$\lambda_* = 2 - 3^{\frac{1}{2}} = 0.2679...$$

This means that the complexity of Newton's method for self-concordant functions is entirely characterized by the parameter a (not dimension n, etc...). Furthermore, in general, one can show $a = O(n)$.

2.5 Homotopy

Suppose now that we want to solve the following program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{s.t.} & f_i(x) \leq 0, \quad i = 1, ..., m \end{array}$$

with $f_i(x) : \mathbb{R}^n \to \mathbb{R}$, convex, strongly self-concordant for $i = 1, ..., n$. For $t > 0$, we replace it by the following unconstrained problem

$$\begin{array}{ll} \text{minimize} & tc^T x + \left(\sum_{i=1}^n f_i(x)\right) \\ & x \in \mathbb{R}^n \end{array}$$

and the complexity analysis can be extended to constrained problems.

2.6 Examples

- Quadratic functions of course...

- Logarithmic barriers for symmetric cones

LP:	$-\sum_{i=1}^{n} \ln x_i$
Second order:	$-\ln\left(y - x^T x\right)$
Semidefinite:	$-\ln\det X$

- The function $x \ln x - \ln x$

- The function $x \ln x$ and its conjugate $\ln\left(\sum_{i=1}^{n} \exp(x_i)\right)$, with small mods... (a.k.a. geometric programming)

2.7 Combination rules

- Affine invariance.

- Stability under convex combination.

- In fact, most operations that preserve convexity also preserve self-concordance.

- A function is self-concordant iff it is self-concordant along all lines.

- If f is strongly self-concordant, then so is it's conjugate f^*.

2.8 Efficient algorithms

- Solve these convex problems with *known complexity bounds*.

- Solve both primal and dual at the same time, hence produce a *certificate* of either *optimality* or *infeasibility*.

- Vast expressive power...

- Reliability similar to LP solvers.

- No "fudging" involved...

3 Symmetric cone programs

- In the following, focus on symmetric cone programs:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax - b \in \mathcal{K} \end{array}$$

- $\mathcal{K}$ is a product of symmetric cones: $\mathcal{K} = LP \times SO^k \times SDP^l$ with

$$\begin{array}{ll} \text{LP:} & \{x \in \mathbb{R}^n : x \geq 0\} \\ \text{Second order:} & \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : \|x\| \leq y\} \\ \text{Semidefinite:} & \{X \in \mathbf{S}^n : X \succeq 0\} \end{array}$$

- Extremely large catalog of applications.

3.1 Classic format

- All the programs that follow are particular instances of:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & \|B_k x + d_k\| \leq C_k x + e_k \quad \text{for } k = 1, \ldots, K \\ & \sum_{j=1}^{n} D_{l,j} x \preceq D_{l,0} \quad \text{for } l = 1, \ldots, L \end{array}$$

where $D_{l,j} \in \mathbf{S}^n$ and $A \preceq B$ means $B - A$ positive semidefinite.

3.2 Example: Robust linear programming

- Solve

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i \quad \text{for } i = 1, \ldots, m \end{array}$$

with a_i in a confidence ellipsoid:

$$\mathcal{E}_i = \{\bar{a}_i + V_i u : \|u\| \leq 1\}.$$

- Find a *robust solution*, a solution valid for all values of a_i:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \sup_{a_i \in \mathcal{E}_i} \left\{ a_i^T x \right\} \leq b_i \quad \text{for } i = 1, \ldots, m \end{array}$$

This is a second order cone program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \left\| V_i^T x \right\| \leq b_i \quad \text{for } i = 1, \ldots, m \end{array}$$

3.3 Stochastic LP

- A similar program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & P\left(a_i^T x \leq b_i\right) \geq \eta \quad \text{for } i = 1, \ldots, m \end{array}$$

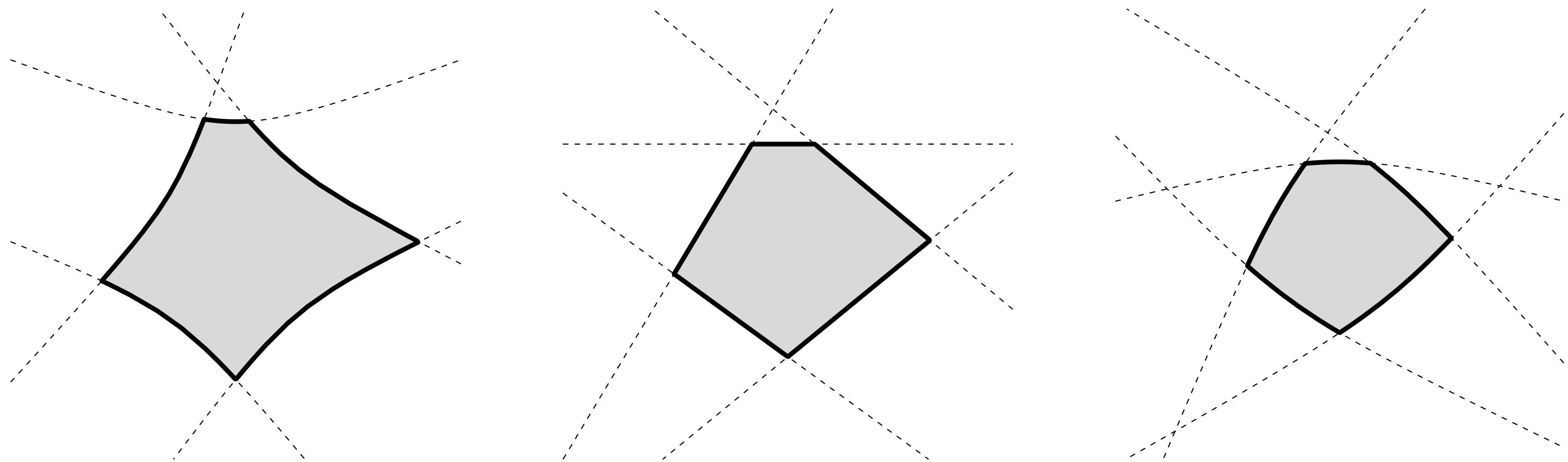
- Suppose $a_i \sim \mathcal{N}(\bar{a}_i, V_i)$.

- The problem becomes:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \Phi^{-1}(\eta) \left\| V_i^{1/2} x \right\| \leq b_i \quad \text{for } i = 1, \ldots, m \end{array}$$

where Φ is the Gaussian CDF.

- In this case, the feasible sets can become non-convex:



$\eta = 10\%$ $\eta = 50\%$ $\eta = 90\%$

3.4 Gamma management

- Following Douady (1995), suppose that we hold a delta hedged portfolio on n assets S_i with gamma Γ

- We want to make it gamma positive

- For liquidity reasons, we can only use options on each individual asset S_i, with gamma given by γ_i (no baskets).

- If delta neutrality is maintained at all times, the gamma positivity condition becomes:

$$\Gamma + \mathbf{diag}\,(x_i \gamma_i) \succeq 0$$

where x_i is the number of options on asset S_i.

- With proportional transaction costs k_i, the cheapest gamma positive portfolio is found by solving

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} k_i \, |x_i| \\ \text{subject to} & \Gamma + \mathbf{diag}\,(x_i \gamma_i) \succeq 0 \quad \text{for } i = 1, ..., n \end{array}$$

which is a *semidefinite program*.

3.5 Libor market model calibration

- Swaption prices can be approximated by:

$$Swaption = level_t \times BS(swap_t, V_T, T)$$

with

$$\begin{aligned} V_T &= \int_t^T \left\| \sum_{i=1}^N \hat{\omega}_i(t) \gamma(s, T_i - s) \right\|^2 ds \\ &= \int_t^T \left(\sum_{i=1}^N \sum_{j=1}^N \hat{\omega}_i(t) \hat{\omega}_j(t) \left\langle \gamma(s, T_i - s), \gamma(s, T_j - s) \right\rangle \right) ds \\ &= \int_t^T \mathbf{Tr}\left(\Omega_t \Gamma_s\right) ds \end{aligned}$$

where $\Gamma_s = \left(\left\langle \gamma(s, T_i - s), \gamma(s, T_j - s) \right\rangle \right)_{i,j}$

- Approximates a sum of lognormals by a lognormal, matching moments.

- The weights $\hat{\omega}_i(t)$ are computed from:

$$\hat{\omega}_i(t) = \omega_i(t)\frac{K(t,T_i)}{swap(t,T,T_N)}$$

where $\omega_i(t)$ are the coefficients in the swap's decomposition as a basket of forwards:

$$swap(t,T,T_n) = \sum_{i=i_T}^{n} \omega_i(t) K(t,T_i^{float})$$

with the weights given by:

$$\omega_i(t) = \frac{coverage(T_i^{float}, T_{i+1}^{float}) B(t, T_{i+1}^{float})}{Level(t, T^{fixed}, T_n^{fixed})}$$

where $0 \leq \omega_i(t) \leq 1$.

3.6 Calibration program

- The calibration problem becomes:

$$\begin{array}{ll} \text{find} & X \\ \text{such that} & \mathbf{Tr}\left(\Omega_i X\right) = \sigma^2_{market,i} T_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

which is a *semidefinite feasibility problem*.

3.7 Objectives

- *Tikhonov regularization* (see Cont (2001) on volatility surface):

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|X\| \leq t \\ & \mathbf{Tr}\left(\Omega_i X\right) = \sigma^2_{market,i} T_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

- *Smoothness*:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\Delta X\| \leq t \\ & \mathbf{Tr}\left(\Omega_i X\right) = \sigma^2_{market,i} T_i \quad \text{for } i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

- *Distance* to a given matrix C:

$$\begin{array}{ll}\text{minimize} & t\\ \text{subject to} & -tI \preceq X - C \preceq tI\\ & \mathbf{Tr}\,(\Omega_i X) = \sigma^2_{market,i} T_i \quad \text{for } i = 1, ..., m\\ & X \succeq 0\end{array}$$

- *Bounds* on the price of another swaption:

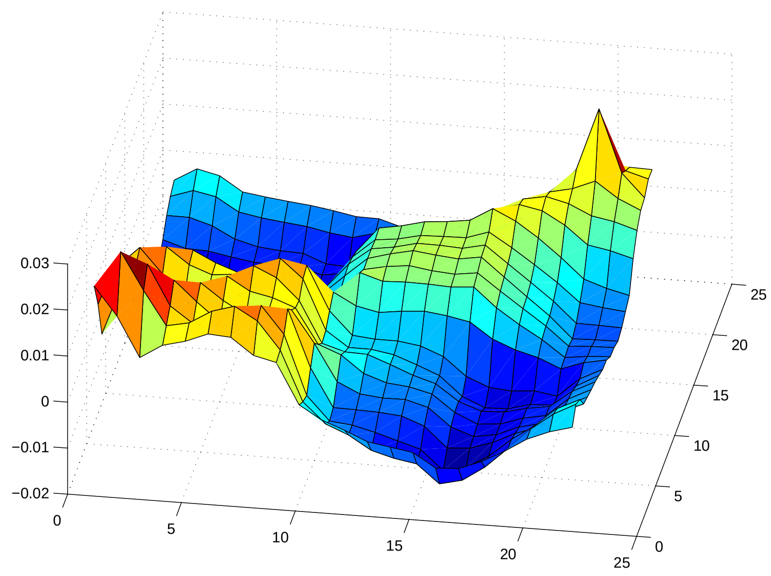
$$\begin{array}{ll}\text{min/max} & \mathbf{Tr}\,(\Omega_0 X)\\ \text{subject to} & \mathbf{Tr}\,(\Omega_i X) = \sigma^2_{market,i} T_i \quad \text{for } i = 1, ..., m\\ & X \succeq 0\end{array}$$

- *Robust* solution (solution centering):

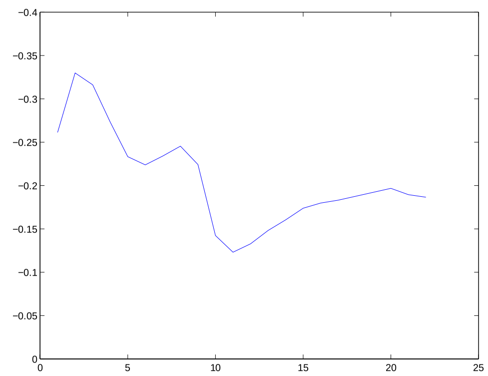
$$\begin{array}{ll}\text{maximize} & t\\ \text{subject to} & \sigma_{Bid,i} T_i + t \leq \mathbf{Tr}\,(\Omega_i X) \leq \sigma_{Ask,i} T_i - t \quad \text{for } i = 1, ..., m\\ & X \succeq tI\\ & t \geq 0\end{array}$$

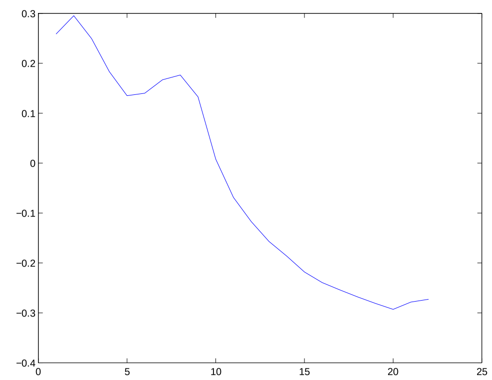
- Or a mix....

- Caveat: $\mathbf{Rank}(X)$. The Minimum rank problem is NP-Complete, but excellent heuristics exist (see Boyd, Fazel & Hindi (2000)).



Smooth calibrated matrix





Level

Spread

3.8 Infeasibility

- If the program is not feasible, we get a Farkas type certificate:

$$\lambda \in \mathbb{R}^M : 0 \preceq \sum_{k=1}^{M} \lambda_k \Omega_k \quad \text{et} \quad \lambda^T \left(\sigma^2 T\right) < 0$$

- This detects an arbitrage: the options with variance $\sigma_k^2 T_k$ with $\lambda_k > 0$ cannot constitute a viable price system within the model.

- Detecting the smallest set of products that admits an arbitrage is NP-complete (MINCARD), but same heuristics apply (see Boyd et al. (2000)).

3.9 Bounds on swaption prices

- The objective can be the BS variance of another swaption.(replicating a particular swaption with more liquid ones):

$$\begin{array}{ll} \text{maximiser} & \sigma^2_{\max} T = \mathbf{Tr}(\Omega_0 X) \\ \text{s.t.} & \mathbf{Tr}(\Omega_k X) = \sigma_k^2 T_k \text{ for } k = 1, \ldots, M \\ & X \succeq 0 \end{array}$$

- The dual program can be interpreted as a hedging program à la Avellaneda & Paras (1996).

- If $BS_k(v)$, is the Black Scholes price of swaption k for a variance v:

$$P = \inf_{\lambda} \left\{ \sum_{k=1}^{M} \lambda_k C_k + \sup_{X \succeq 0} \left(BS_0(\mathbf{Tr}(\Omega_0 X)) - \sum_{k=1}^{M} \lambda_k BS_k\left(\mathbf{Tr}(\Omega_k X)\right) \right) \right\}$$

or again

$$\text{Price} = \text{Min}\left\{\text{PV static hedge} + \text{Max}\left(\text{PV residual}\right)\right\}$$

- Example on a Nov. 6 2000 dataset. Calibrated using all caplets and the following swaptions: 5Y into 5Y, 5Y into 2Y, 5Y into 10Y, 2Y into 2Y, 2Y into 5Y, 7Y into 5Y, 10Y into 5Y, 10Y into 2Y, 10Y into 10Y, 7Y into 3Y, 4Y into 6Y, 17Y into 3Y. (Figure 1) (Data courtesy of BNP Paribas, Londres).

- The model used here is extremely simple (stationary in sliding Libor) but it gives reasonable bounds for short maturities.

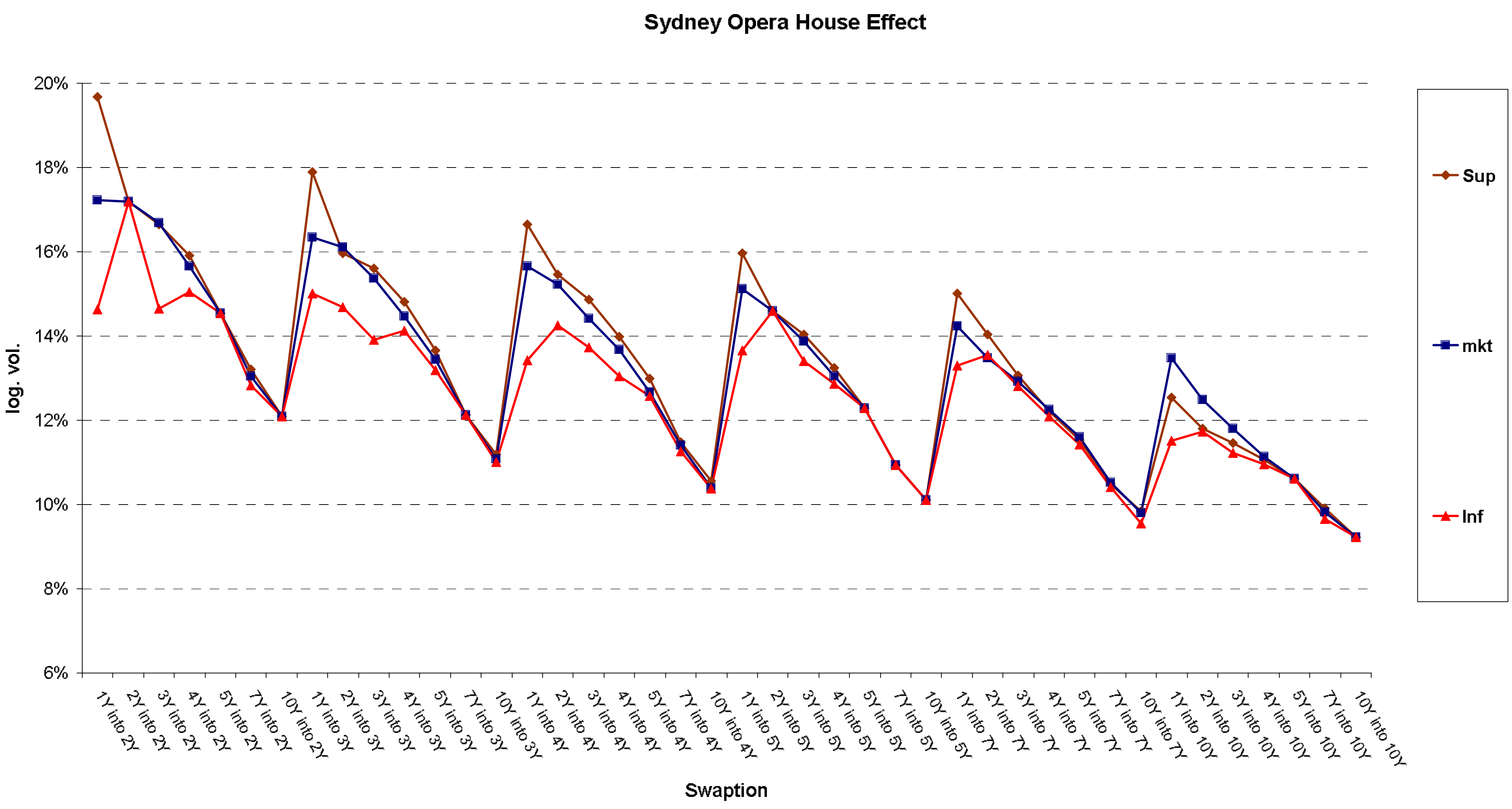


Figure 1: Bornes inf. et sup. sur le prix des swaptions.

3.10 Robust portfolio allocation with uncertain data

- Following El Ghaoui (1999), in a one period model. Assets p_i, for $i = 1, ..., n$, with mean p and covariance Σ.

- Partial information on Σ, i.e. $\Sigma \in \mathcal{U}$ where

$$\mathcal{U} = X \in \mathbf{S}_+^n : \quad \begin{array}{ll} X_{i,j} \geq 0 & (i,j) \in I_+ \\ X_{i,j} \leq 0 & (i,j) \in I_- \\ X_{i,j} = \Sigma_{i,j}^0 & (i,j) \in I_0 \end{array}$$

- Set of admissible portfolios given by $Ax \leq b$.

- Objective: minimize the worst-case variance:

$$\begin{array}{lll} \text{minimize} & \max & x^T \Sigma x \\ & \text{s.t.} & \Sigma \in \mathcal{U} \\ & & Ax \leq b \end{array}$$

- Solution given by the following semidefinite program:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}\left(X\Sigma^0\right) \\ \text{subject to} & X \succeq xx^T \\ & Ax \leq b \\ & X \in \mathcal{U} \end{array}$$

or explicitly:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}\left(X\Sigma^0\right) \\ \text{subject to} & Ax \leq b \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0 \\ & X_{i,j} \geq 0 \quad (i,j) \in I_+ \\ & X_{i,j} \leq 0 \quad (i,j) \in I_- \\ & X_{i,j} = \Sigma^0_{i,j} \quad (i,j) \in I_0 \\ & X \succeq 0 \end{array}$$

- The optimal portfolio is then given by x^{opt}.

3.11 The Hamburger moment problem

Exact solution via semidefinite programming: $y = (y_0, y_1, ..., y_{2m})$ is a moment sequence iff the corresponding Hankel matrix is positive semidefinite:

$$H_m(y) = \begin{bmatrix} y_0 & y_1 & y_2 & \cdot & y_m \\ y_1 & y_2 & \cdot & \cdot & y_{m+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ y_m & y_{m+1} & \cdot & y_{2m-1} & y_{2m} \end{bmatrix} \succeq 0.$$

Let μ be the corresponding measure with $y_i = \int x^i d\mu$.

3.12 Dual: sum of squares polynomials

Hilberth's 17^{th} problem: let $p(x) \in \mathbb{R}[x]$ (dimension one):

$$p(x) \geq 0 \Leftrightarrow p(x) = \sum_{i=1}^{r} q_i(x)^2$$

Again, solution via semidefinite programming, $p(x) \geq 0$ for $x \in \mathbb{R}$ iff

$$\begin{array}{rl} p(x) = & \mathbf{Tr}\,(X H_m(y)) \quad \text{for } x \in \mathbb{R} \\ & X \succeq 0 \end{array}$$

where $y = (1, x, ..., x^{2m})$.

3.13 Conic duality

- On $\mathbb{R}^n$ the situation is different: see e.g. Berg (1980):

$$s \text{ is p.s.d.} \Leftrightarrow \langle s, p_\alpha \rangle \geq 0, \forall\, p(x) \in \mathbb{R}^n[x] \text{ with } p(x)\ SOS$$

and

$$s \text{ is a moment sequence} \Leftrightarrow \langle s, p_\alpha \rangle \geq 0, \forall\, p(x) \in \mathbb{R}^n[x] \text{ with } p(x) \geq 0.$$

- See Putinar (1993) and Lasserre (2001) on the solution to the $\mathbb{K}$-moment problem by SOS polynomials and semidefinite programming.

3.14 Software

- SEDUMI (GPL license), for symmetric cone programs.

  `http://fewcal.kub.nl/sturm/software/sedumi.html`

- MOSEK (Free for academic use), for general convex programs and 0-1 programs.

  `http://www.mosek.com`

4 Conclusion

- Up to now, said X times the word ”convex”, said ”nonlinear” only twice (here included).

Nonlinearity is irrelevant to computational complexity

- Very consitent theory to describe computational complexity of a large class of convex problems

- In practice (for small sizes): experience comparable to that of linear progamming. Fast reliable solvers you can forget...

References

Avellaneda, M. & Paras, A. (1996), 'Managing the volatility risk of portfolios of derivative securities: the lagrangian uncertain volatility model', *Applied Mathematical Finance* 3, 21–52.

Ben-Tal, A. & Nemirovski, A. (2001), *Lectures on modern convex optimization : analysis, algorithms, and engineering applications*, MPS-SIAM series on optimization, Society for Industrial and Applied Mathematics : Mathematical Programming Society, Philadelphia, PA.

Berg, C. (1980), The multidimensional moment problem and semi-groups, *in* H. Landau, ed., 'Moments in Mathematics', AMS, Providence, RI, pp. 110–124.

Boyd, S. P., Fazel, M. & Hindi, H. (2000), 'A rank minimization heuristic with application to minimum order system approximation.', *Working paper. American Control Conference, September 2000* .

Boyd, S. & Vandenberghe, L. (2003), *Convex Optimization*, To appear.

Cont, R. (2001), 'Inverse problems in financial modeling: theoretical and numerical aspects of model calibration.', *Lecture Notes, Princeton University.* .

Douady, R. (1995), 'Optimisation du gamma d'une option sur panier ou sur spread en l'absence d'options croisées.', *Working paper* .

El Ghaoui, L. (1999), 'Robust portfolio allocation with uncertain data', *mimeo* .

Karmarkar, N. K. (1984), 'A new polynomial-time algorithm for linear programming', *Combinatorica* 4, 373–395.

Khachiyan, L. G. (1979), 'A polynomial algorithm in linear programming (in Russian)', *Doklady Akademiia Nauk SSSR* 224, 1093–1096.

Lasserre, J. B. (2001), 'Global optimization with polynomials and the problem of moments', *SIAM Journal on Optimization* 11(3), 796–817.

Nemirovskii, A. S. & Yudin, D. B. (1979), 'Problem complexity and method efficiency in optimization', *Nauka (published in English by John Wiley, Chichester, 1983)* .

Nesterov, I. & Nemirovskii, A. (1994), *Interior-point polynomial algorithms in convex programming*, Society for Industrial and Applied Mathematics, Philadelphia.

Putinar, M. (1993), 'Positive polynomials on compact semi-algebraic sets', *Indiana University Mathematics Journal* 42(3), 969–984.