Nonconvex optimization under the hood

Robert Luce

May 2022

GUROBI OPTIMIZATION

Nonconvex optimization under the hood

Robert Luce

May 2022

Cuts

Cut

Setup

We consider the problem

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & x^T Q_0 x + c^T x \\
\text{s.t.} \quad & Ax = b \\
& x^T Q_k x + p_k^T x \leq d_k \\
& l \leq x \leq u \\
& x_{\mathcal{I}} \in \mathbb{Z}
\end{aligned}
$$

with all $Q_k \in \mathbb{R}^{n \times n}$ symmetric.

- Our goal: find a provably global optimal solution.
- Our solution strategy: Branch-and-bound.

Simplified setup

We consider the problem

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & x^T Q x + c^T x \\
\text{s.t.} \quad & Ax = b \\
& x \geq 0 \\
& x_{\mathcal{I}} \in \mathbb{Z}
\end{aligned}
$$

- We are interested in the case where $x^T Q x$ is nonconvex.
- Problem: Relaxing $x_{\mathcal{I}} \in \mathbb{Z}$ gives us only a *nonconvex* continuous problem.
- Need to fix this first to make BnB effective!

GUROBI OPTIMIZATION

Extended formulation & McCormick relaxation

Basic idea:

- For each appearing quadratic term $x_i x_j$ introduce an auxiliary variable X_{ij}.
- Add some polyhedral constraints $(x, X) \in \mathcal{S}$ that connect $x_i x_j$ with X_{ij} (linear envelope of $x_i x_j$).
- The envelope becomes tighter in the course of branching, bound changes for x_i, x_j propagate to bound changes for X_{ij}.

Challange: We may need to branch many times until the relaxation solution satisfies

$$xx^T = X.$$

GUROBI OPTIMIZATION

Cuts from SDP outer approximation 1

We will use the $xx^T = X$ to derive globally valid cutting planes for the relaxed extended formulation.

Cuts from SDP outer approximation 1

We will use the $xx^T = X$ to derive globally valid cutting planes for the relaxed extended formulation.
For any $x \in \mathbb{R}^n, X \in \mathbb{R}^{n\times n}$ We have

$$
\begin{aligned}
xx^T = X \Rightarrow xx^T &\preccurlyeq X \\
&\Leftrightarrow 0 \preccurlyeq X - xx^T \\
&\Leftrightarrow 0 \preccurlyeq \begin{bmatrix} 1 & 0 \\ 0 & X - xx^T \end{bmatrix} \\
&\Leftrightarrow 0 \preccurlyeq \begin{bmatrix} 1 & 0 \\ x & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & X - xx^T \end{bmatrix} \begin{bmatrix} 1 & x^T \\ 0 & I \end{bmatrix} \\
&\Leftrightarrow 0 \preccurlyeq \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} =: \hat{X}
\end{aligned}
$$

How do we derive cuts from $0 \preccurlyeq \hat{X}$?

GUROBI OPTIMIZATION

Cuts from outer approximation 2

Recall

$$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} =: \hat{X}$$

From the variational characterization

$$\hat{X} \succcurlyeq 0 \;\Leftrightarrow\; v^T \hat{X} v \geq 0 \quad \forall\, v \in \mathbb{R}^n$$

we see that a solution (x^*, X^*) for the relaxation is cut off by the *linear* cutting plane $v^T \hat{X} v \geq 0$ by any $v \in \mathbb{R}^n$ satisfying

$$v^T \hat{X}^* v < 0.$$

GUROBI OPTIMIZATION

Characterization of cut-defining vectors

- Let (λ, v) be a normalized eigenpair with $\lambda < 0$, then
$$v^T \hat{X}^* v = \lambda v^T v = \lambda < 0.$$
- More generally, let $\mathcal{U} := \text{span}\{v_1, \dots, v_s\}$ be the subspace generated from eigenvectors corresponding to all negative eigenvalues. Then any $v \in \mathcal{U}$ defines a cut.
- Reverse: *any* cut-defining v satisfies $\text{proj}_{\mathcal{U}}(v) \neq 0$
- Even better: If $v \notin \mathcal{U}$, and $w = \text{proj}_{\mathcal{U}}(v)$, then $w^T \hat{X}^* w \leq v^T \hat{X} v$.

Characterization of cut-defining vectors

- Let (λ, v) be a normalized eigenpair with $\lambda < 0$, then
$$v^T \hat{X}^* v = \lambda v^T v = \lambda < 0.$$
- More generally, let $\mathcal{U} := \text{span}\{v_1, \dots, v_s\}$ be the subspace generated from eigenvectors corresponding to all negative eigenvalues. Then any $v \in \mathcal{U}$ defines a cut.
- Reverse: *any* cut-defining v satisfies $\text{proj}_{\mathcal{U}}(v) \neq 0$
- Even better: If $v \notin \mathcal{U}$, and $w = \text{proj}_{\mathcal{U}}(v)$, then $w^T \hat{X}^* w \leq v^T \hat{X} v$.

Conclusion: $\mathcal{U}$ is the right place to look for cuts.

GUROBI OPTIMIZATION

Characterization of cut-defining vectors

- Let (λ, v) be a normalized eigenpair with $\lambda < 0$, then

$$v^T \hat{X}^* v = \lambda v^T v = \lambda < 0.$$

- More generally, let $\mathcal{U} := \text{span}\{v_1, \ldots, v_s\}$ be the subspace generated from eigenvectors corresponding to all negative eigenvalues. Then any $v \in \mathcal{U}$ defines a cut.
- Reverse: *any* cut-defining v satisfies $\text{proj}_{\mathcal{U}}(v) \neq 0$
- Even better: If $v \notin \mathcal{U}$, and $w = \text{proj}_{\mathcal{U}}(v)$, then $w^T \hat{X}^* w \leq v^T \hat{X} v$.

Conclusion: $\mathcal{U}$ is the right place to look for cuts.

Problems: $\mathcal{U}$ is expensive to compute for large n, and the number of nonzeros in the cut are $\frac{n(n+1)}{2} + n$.

GUROBI OPTIMIZATION

Cuts from submatrices

For $\mathcal{I} \subseteq [n]$ we define the submatrix of $\hat{X}$ induced by $\mathcal{I}$ by

$$\hat{X}_{\mathcal{I}} := \begin{bmatrix} 1 & x(\mathcal{I})^T \\ x(\mathcal{I}) & X(\mathcal{I},\mathcal{I}) \end{bmatrix}.$$

Passing to subsets is a way around computational burden, but since

$$\min_{v \in \mathbb{R}^n} v^T \hat{X} v \le \min_{v \in \mathrm{span}\{e_i\}_{i \in \mathcal{I}}} v^T \hat{X} v = \min_{v \in \mathbb{R}^{|\mathcal{I}|}} v^T \hat{X}_{\mathcal{I}} v$$

a cut may be quite a bit weaker than the best possible cut on $\hat{X}$.

Sparse extended formulations

Typically we will not add *all* the variables X_{ij} in our extended formulation. For simplicity assume that we have added all variables corresponding to the incidence graph $G_Q = (V, E) := G(Q)$ though.

GUROBI OPTIMIZATION

Sparse extended formulations

Typically we will not add *all* the variables X_{ij} in our extended formulation. For simplicity assume that we have added all variables corresponding to the incidence graph $G_Q = (V, E) := G(Q)$ though.

Simple heuristic 1:

- Pick any "small" clique $\mathcal{C}$ in G_Q.
- Apply cut heuristic to $G_Q[\mathcal{C}]$.

Sparse extended formulations

Typically we will not add *all* the variables X_{ij} in our extended formulation. For simplicity assume that we have added all variables corresponding to the incidence graph $G_Q = (V, E) := G(Q)$ though.

Simple heuristic 1:

- Pick any "small" clique $\mathcal{C}$ in G_Q.
- Apply cut heuristic to $G_Q[\mathcal{C}]$.

Simple heuristic 2:

- Compute a chordal completion C of G_Q.
- For each maximal clique of C (that is still small enough...) fill entries in X^* by

$$[X^*]_{ij} = \begin{cases} X^*_{ij} & \text{if } (i,j) \in E \\ x^*_i x^*_j & \text{otherwise,} \end{cases}$$

  and relax "missing" variables in the cut by an upper bound.
- If cut still cuts off (x^*, X^*), take it!

GUROBI OPTIMIZATION

Eigenspace guided submatrix selection

Now consider the setting where G_Q is large and sparse. We can compute an s-dimensional approximation to $\mathcal{U}$ (e.g., Lanczos, Krylov-Schur).

- Basic operation: Matrix vector products with $\hat{X}^*$, cost $\mathcal{O}\left(n + |E|\right)$ each, and a few eigensolves of size s.
- *If the method converges*, we obtain a $U \in \mathbb{R}^{n,s}$ with orthonormal columns, such that $\text{span}(U) \subseteq \mathcal{U}$. (Or a certificate that no cuts can be separated.)

GUROBI OPTIMIZATION

Eigenspace guided submatrix selection

Now consider the setting where G_Q is large and sparse. We can compute an s-dimensional approximation to $\mathcal{U}$ (e.g., Lanczos, Krylov-Schur).

- Basic operation: Matrix vector products with $\hat{X}^*$, cost $\mathcal{O}(n + |E|)$ each, and a few eigensolves of size s.
- *If the method converges*, we obtain a $U \in \mathbb{R}^{n,s}$ with orthonormal columns, such that $\text{span}(U) \subseteq \mathcal{U}$. (Or a certificate that no cuts can be separated.)

With U at hand, we can:

1. Generate dense cuts as before.
2. Project U on a selection matrix, i.e., find a matrix

$$P = \left[e_1, e_{i_1}, \ldots, e_{i_r}\right], \in \mathbb{R}^{n,r} \quad r \leq s$$

such that $\|U - P\|$ is (somewhat) small, and separate a cut on $P^T \hat{X}^* P = \hat{X}^*_{\mathcal{I}}$.

Heuristics

Heuristic

Simplified problem setting

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x \geq 0 \end{aligned}$$

Because our general problem setting contains only linear and quadratic constraints, both f and c are trivially twice differentiable, and $\nabla^2 f$ and all of $\nabla^2 c_i$ are Lipschitz continuous.

First order (FO) optimality conditions at optimum (x^*, y^*, z^*)

$$
\begin{aligned}
\nabla f(x^*) + \nabla c(x^*) y^* - z^* &= 0 \\
c(x^*) &= 0 \\
0 \leq z^* \perp x^* &\geq 0
\end{aligned}
$$

- These do *not* guarantee a local optimum.
- A few other optimality measures need to be considered.
- Not all are actually computable or even heuristically assessable.

GUROBI OPTIMIZATION

Basic ingredients

In order to solve this problem with an iterative scheme, we need to

1. have a device to deal with complementary conditions (nonsmooth!),
2. find directions of local "improvement", and
3. ensure global convergence.

GUROBI OPTIMIZATION

Basic ingredients

In order to solve this problem with an iterative scheme, we need to

1. have a device to deal with complementary conditions (nonsmooth!),
2. find directions of local "improvement", and
3. ensure global convergence.

Ingredients for addressing these:

1. Homotopy method (aka barrier function)
2. Newton method
3. Line search, filter, feasibility relaxation

Homotopy on FO KKT system

We replace condition $z \perp x$ by a *sequence* of constraints

$$\operatorname{diag}(x)z =: Xz = \mu\mathbf{1},$$

with parameter $\mu \to 0$. Thus we end up with a sequence of nonlinear systems

$$\begin{aligned} \nabla f(x) + \nabla c(x)y - z &= 0 \\ c(x) &= 0 \\ Xz &= \mu\mathbf{1} \\ x, z &\geq 0 \end{aligned}$$

whose solutions approach a solution of original FO KKT system.

GUROBI OPTIMIZATION

Homotopy on FO KKT system

We replace condition $z \perp x$ by a *sequence* of constraints

$$\mathrm{diag}(x)z =: Xz = \mu\mathbf{1},$$

with parameter $\mu \to 0$. Thus we end up with a sequence of nonlinear systems

$$\begin{aligned} \nabla f(x) + \nabla c(x)y - z &= 0 \\ c(x) &= 0 \\ Xz &= \mu\mathbf{1} \\ x, z &\geq 0 \end{aligned}$$

whose solutions approach a solution of original FO KKT system.

- Imposed regularity on f, c enters analysis of homotopy path.
- Additional convergence conditions: LICQ, strict complementarity, Hessian uniformly bounded from below, nonempty interior, ...
- Optima $x^*(\mu)$ are guaranteed to converge only in a neighborhood of 0.

GUROBI OPTIMIZATION

Newton method

Basic Newton iteration for a function $f : \mathbb{R}^n \supset D \to \mathbb{R}^n$: $x_{k+1} = x_k - \nabla f(x_k)^{-1} f(x_k)$.

GUROBI OPTIMIZATION

Newton method

Basic Newton iteration for a function $f : \mathbb{R}^n \supset D \to \mathbb{R}^n$: $x_{k+1} = x_k - \nabla f(x_k)^{-1} f(x_k)$.
Applying Newton method to μ-FO systems:

$$\begin{bmatrix} \nabla^2 f(x_k) + \sum_i y_i \nabla^2 c_i(x_k) + X^{-1}Z & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) - \nabla c(x_k) y_k + \mu X^{-1}\mathbf{1} \\ -c(x) \end{bmatrix}$$

Newton method

Basic Newton iteration for a function $f : \mathbb{R}^n \supset D \to \mathbb{R}^n$: $x_{k+1} = x_k - \nabla f(x_k)^{-1} f(x_k)$.
Applying Newton method to μ-FO systems:

$$\begin{bmatrix} \nabla^2 f(x_k) + \sum_i y_i \nabla^2 c_i(x_k) + X^{-1}Z & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) - \nabla c(x_k) y_k + \mu X^{-1}\mathbf{1} \\ -c(x) \end{bmatrix}$$

- Newton directions improve feasibility *of the FO system* – but possibly not of any second order, or other sufficient optimality conditions.
- Need to apply heuristics to get actual "improving" direction from the Newton scheme.
- Need to damp the Newton steps to ensure nonnegativity.

Global convergence

Basic problem: Feasible region is nonconvex, how do we guarantee convergence to a local optimum?

Global convergence

Basic problem: Feasible region is nonconvex, how do we guarantee convergence to a local optimum?

1. Use line search for Newton directions. Cut back on step length until new point is an "improvement" by some metrics.
2. Use "filter" to forbid steps into already dominated regions.
3. Use feasibility relaxation if stuck at a point, i.e., solve

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|p\|_1 + \|x - x_k\|_2 \\ \text{s.t.} \quad & c(x) + p = 0 \\ & x \geq 0 \end{aligned}$$

GUROBI OPTIMIZATION

Conclusions

- Singled-out subproblems lead to other interesting problems!
- Nonconvex global optimization is fun.

Conclusions

- Singled-out subproblems lead to other interesting problems!
- Nonconvex global optimization is fun.

Thanks!

GUROBI OPTIMIZATION