

Optimization Techniques for Geometry Processing

Justin Solomon

Sebastian Claici



Justin Solomon

Sebastian Claici

More Specifically

 $\min_{x \in \mathbb{R}^n} f(x)$ s.t. g(x) = 0 $h(x) \ge 0$

Two Roles

Client

Which optimization tool is relevant?

Designer Can I design an algorithm for this problem?

Our Bias



Optimization is a <u>huge</u> field.

Geometry is Special



Very Large Nonconvex Problem

$$\begin{array}{ll} \min_{\phi:\Sigma \to \mathbb{R}^2} & \text{Distortion}[\phi] \\ & \text{s.t.} & \phi \text{ is one-to-one} \end{array}$$

Difficult because:

- Distortion is nonlinear/nonconvex
- Parameterization *must* be one-to-one



What's Wrong with Gradient Descent?



Part I (Justin)

Vocabulary

Simple examples

Unconstrained optimization

Equality-constrained optimization

Part II (Sebastian)

Inequality constraints

Advanced algorithms

Examples

Conclusion

Part I (Justin)

Vocabulary (basic material!)

Simple examples

Unconstrained optimization

Equality-constrained optimization

 $\min_{x \in \mathbb{R}^n} f(x)$ s.t. g(x) = 0 $h(x) \ge 0$

Objective ("Energy Function")

$$\min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t.} \ g(x) = 0 \\ h(x) \ge 0$$

Equality Constraints

 $\min_{x \in \mathbb{R}^n} f(x)$ s.t. g(x) = 0 $h(x) \ge 0$

Inequality Constraints

$$f: \mathbb{R}^n \to \mathbb{R}$$

$$\to \nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right)$$

Gradient

https://en.wikipedia.org/?title=Gradient



https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

Jacobian

$$f: \mathbb{R}^n \to \mathbb{R} \to H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$



$$f(x) \approx f(x_0) + \nabla f(x_0)^\top (x - x_0) + (x - x_0)^\top H f(x_0) (x - x_0)$$

http://math.etsu.edu/multicalc/prealpha/Chap2/Chap2-5/10-3a-t3.gif



 $\nabla f(x)$

(unconstrained)

Local max

f(x)• X

Saddle point

Critical point

Local min

Bonus: Multi-Objective Optimization



Design Space Pareto Set

Performance Space Pareto Front

Roughly: $\alpha \mapsto \min_x \sum_k \alpha_k f_k(x)$

Example



Image from "Interactive Exploration of Design Trade-Offs" Schulz, Wang, Grinspun, Solomon, & Matusik

Part I (Justin)

Vocabulary

Simple examples

Unconstrained optimization

Equality-constrained optimization

Encapsulates Many Problems

$$\min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } g(x) = 0 \\ h(x) \ge 0$$

$$Ax = b \leftrightarrow f(x) = \|Ax - b\|_2$$

 $Ax = \lambda x \leftrightarrow f(x) = ||Ax||_2, g(x) = ||x||_2 - 1$

Roots of $g(x) \leftrightarrow f(x) = 0$

How effective are generic optimization tools?



How effective are generic optimization tools?



Generic Advice

Try the simplest solver first.

Quadratic with Linear Equality

$$\begin{array}{ccc} \min_{x} & \frac{1}{2}x^{\top}Ax - b^{\top}x + c \\ \text{s.t.} & Mx = v \\ \text{(assume A is symmetric and positive definite)} \\ & \downarrow \\ A & M^{\top} \\ M & 0 \end{array} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ v \end{pmatrix}$$

Special Case: Least-Squares

$$\min_{x} \frac{1}{2} \|Ax - b\|_{2}^{2}$$

$$\rightarrow \min_{x} \frac{1}{2} x^{\top} A^{\top} A x - b^{\top} A x + \|b\|_{2}^{2}$$

$$\implies A^{\top}Ax = A^{\top}b$$

Normal equations (better solvers for this case!)

Example: Mesh Embedding



G. Peyré, mesh processing course slides

Linear Solve for Embedding

$$\min_{\substack{x_1, \dots, x_{|V|} \\ \text{s.t.} \quad x_v \text{ fixed } \forall v \in V_0 } } \sum_{ij \in E} w_{ij} \|x_i - x_j\|_2^2$$

w_{ij} ≡ 1: Tutte embedding
 w_{ij} from mesh: Harmonic embedding

Assumption: *w* symmetric.

Linear Solver Considerations

- Never construct A⁻¹ explicitly (if you can avoid it)
- Added structure helps
 <u>Sparsity</u>, symmetry, positive definiteness

$inv(A)*b \ll (A'*A) \setminus (A'*b) \ll A \setminus b$

Two Classes of Solvers

Direct (explicit matrix)

- Dense: Gaussian elimination/LU, QR for least-squares
- Sparse: Reordering (SuiteSparse, Eigen)

Iterative (*αpply* matrix repeatedly)

- Positive definite: Conjugate gradients
- Symmetric: MINRES, GMRES
- Generic: LSQR

Very Common: Sparsity



Returning to Parameterization

$$\min_{\substack{x_1, \dots, x_{|V|} \\ \text{s.t.} \quad x_v \text{ fixed } \forall v \in V_0 } } \sum_{ij \in E} w_{ij} \|x_i - x_j\|_2^2$$

What if
$$V_0 = \{\}$$
?

Back to Parameterization



Mullen et al. "Spectral Conformal Parameterization." SGP 2008.

$$\min_{\substack{u\\ u^{\top}Be=0 \\ u^{\top}Bu=1}} u^{\top} L_C u \quad \longleftrightarrow \quad L_c u = \lambda B u$$

Nontriviality Constraint

$$\left\{\begin{array}{cc} \min_{x} & \|Ax\|_{2} \\ \text{s.t.} & \|x\|_{2} = 1 \end{array}\right\} \mapsto A^{\top}Ax = \lambda x$$

Prevents trivial solution $x \equiv 0$.

Extract the smallest eigenvalue.

Basic Idea of Eigenalgorithms

$$\begin{aligned} A\vec{v} &= c_1 A\vec{x}_1 + \dots + c_n A\vec{x}_n \\ &= c_1 \lambda_1 \vec{x}_1 + \dots + c_n \lambda_n \vec{x}_n \text{ since } A\vec{x}_i = \lambda_i \vec{x}_i \\ &= \lambda_1 \left(c_1 \vec{x}_1 + \frac{\lambda_2}{\lambda_1} c_2 \vec{x}_2 + \dots + \frac{\lambda_n}{\lambda_1} c_n \vec{x}_n \right) \\ A^2 \vec{v} &= \lambda_1^2 \left(c_1 \vec{x}_1 + \left(\frac{\lambda_2}{\lambda_1} \right)^2 c_2 \vec{x}_2 + \dots + \left(\frac{\lambda_n}{\lambda_1} \right)^2 c_n \vec{x}_n \right) \\ &\vdots \\ A^k \vec{v} &= \lambda_1^k \left(c_1 \vec{x}_1 + \left(\frac{\lambda_2}{\lambda_1} \right)^k c_2 \vec{x}_2 + \dots + \left(\frac{\lambda_n}{\lambda_1} \right)^k c_n \vec{x}_n \right). \end{aligned}$$

Combining Tools So Far

Roughly:

1.Extract Laplace-Beltrami eigenfunctions: $L\phi_i = \lambda_i A \phi_i$

2. Find mapping matrix (linear solve!): $\min_{A \in \mathbb{R}^{n \times n}} \|AF_0 - F\|_{\text{Fro}}^2 + \alpha \|A\Delta_0 - \Delta A\|_{\text{Fro}}^2$



Ovsjanikov et al. "Functional Maps." SIGGRAPH 2012.

Part I (Justin)

Vocabulary

Simple examples

Unconstrained optimization

 Equality-constrained optimization

Unconstrained Optimization

Unstructured.



Gradient descent

$$egin{aligned} &\lambda_0 = 0, \lambda_s = rac{1}{2}(1 + \sqrt{1 + 4\lambda_{s-1}^2}), \gamma_s = rac{1 - \lambda_2}{\lambda_{s+1}} \ &y_{s+1} = x_s - rac{1}{eta}
abla f(x_s) \ &x_{s+1} = (1 - \gamma_s) y_{s+1} + \gamma_s y_s \end{aligned}$$

Quadratic convergence on convex problems! (Nesterov 1983)

Accelerated gradient descent

$$x_{k+1} = x_k - \left[Hf(x_k)\right]^{-1} \nabla f(x_k)$$



Newton's Method

$$x_{k+1} = x_k - M_k^{-1} \nabla f(x_k)$$
Hessian
approximation

(Often sparse) approximation from previous samples and gradients
 Inverse in closed form!

Quasi-Newton: BFGS and friends

Recent Trend: Preconditioning



Examples:

- "Accelerated Quadratic Proxy for Geometric Optimization" (Kovalsky et al. 2016)
- "Scalable Locally Injective Maps" (Rabinovich et al. 2016)
- "Isometry-Aware Preconditioning for Mesh Parameterization" (Claici et al. 2017)
- ...several others

Helps in Practice



"Accelerated Quadratic Proxy for Geometric Optimization" (Kovalsky et al. 2016)

Example: Shape Interpolation



Fröhlich and Botsch. "Example-Driven Deformations Based on Discrete Shells." CGF 2011.

Interpolation Pipeline

Roughly:

1. Linearly interpolate edge lengths and dihedral angles.

$$\ell_e^* = (1-t)\ell_e^0 + t\ell_e^1$$
$$\theta_e^* = (1-t)\theta_e^0 + t\theta_e^1$$

2. Nonlinear optimization for vertex positions.

$$\min_{x_1,\dots,x_m} \lambda \sum_e w_e (\ell_e(x) - \ell_e^*)^2$$

Sum of squares: Gauss-Newton

$$+\mu\sum_{e}w_{b}(\theta_{e}(x)-\theta_{e}^{*})^{2}$$

Software

Matlab: fminunc or minfunc C++: libLBFGS, dlib, others

Typically provide functions for function and gradient (and optionally, Hessian).



Part I (Justin)

Vocabulary

Simple examples

Unconstrained optimization

Equality-constrained optimization

Lagrange Multipliers: Idea



Lagrange Multipliers: Idea



Lagrange Multipliers: Idea



Example: Symmetric Eigenvectors

$$f(x) = x^{\top} A x \implies \nabla f(x) = 2Ax$$
$$g(x) = \|x\|_2^2 \implies \nabla g(x) = 2x$$
$$\implies Ax = \lambda x$$

Use of Lagrange Multipliers

Turns constrained optimization into unconstrained root-finding.

$$\nabla f(x) = \lambda \nabla g(x)$$
$$g(x) = 0$$

Many Options

Reparameterization

Eliminate constraints to reduce to unconstrained case

Newton's method

Approximation: quadratic function with linear constraint

Penalty method

Augment objective with barrier term, e.g. $f(x) + \rho |g(x)|$

Trust Region Methods



Example: Levenberg-Marquardt

Example: Polycube Maps



Huang et al. "L1-Based Construction of Polycube Maps from Complex Shapes." TOG 2014.

$$\begin{aligned} & \underset{X \in \mathcal{A}_{b_i}}{\min_X \sum_{b_i}} \quad \mathcal{A}(b_i; X) \| n(b_i; X) \|_1 \\ & \text{s.t.} \quad \sum_{b_i} \mathcal{A}(b_i; X) = \sum_{b_i} \mathcal{A}(b_i; X_0) \end{aligned}$$

Preserve area

Note: Final method includes more terms!

Shielded Line Search



Start with a feasible point Don't ever enter the infeasible region Use a barrier in the objective

Bijective Parameterization with Free Boundaries (Smith & Schaefer, SIGGRAPH 2015)

Convex Optimization Tools



Try lightweight options

Iteratively Reweighted Least Squares

$$\begin{split} \min_{x} \sum_{i} \phi(x \top a_{i} + b_{i}) \leftrightarrow \begin{cases} \min_{x, y_{i}} \sum_{i} y_{i}(x^{\top} a_{i} + b_{i})^{2} \\ \text{s.t. } y_{i} &= \phi(x^{\top} a_{i} + b_{i})(x^{\top} a_{i} + b_{i})^{-2} \end{cases} \\ \\ \overbrace{\overset{x_{i}}{\overset{x}}{\overset{x_{i}}{\overset{x_{i}}{\overset{x_{i}}{\overset{x_{i}}{\overset{x}}{\overset{x_{i}}{\overset{x}}{\overset{x_{i}}{\overset{x}}{\overset{x}{\overset{x}}{\overset{x}}{\overset{x}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}}{\overset{x}{\overset{x}}{\overset{x}}{\overset{x}{\overset{x}}{\overset{x}}{\overset{x}{\overset{x}}{\overset{x}}{\overset{$$

Repeatedly solve linear systems

Alternating Projection



Iterative Shrinkage-Thresholding

$$\begin{aligned} x_{t+1} &= x_t - \eta \nabla f(x_t) \\ \iff x_{t+1} = \arg\min_x \left[f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2\eta} \|x - x_t\|_2^2 \right] \\ \iff x_{t+1} = \arg\min_x \frac{1}{2\eta} \|x - (x_t - \eta \nabla f(x_t))\|_2^2 \end{aligned}$$

Decompose as sum of hard part *f* and easy part *g*.

To minimize
$$f(x) + g(x)$$
:
 $x_{t+1} = \arg \min_{x} \left[g(x) + \frac{1}{2\eta} \left\| x - (x_t - \eta \nabla f(x_t)) \right\|_2^2 \right]$

FISTA combines with Nesterov descent!

https://blogs.princeton.edu/imabandit/2013/04/11/orf523-ista-and-fista/

Augmented Lagrangians

$$\min_{x} f(x) \\ \text{s.t.} g(x) = 0 \\ \downarrow \\ \min_{x} f(x) + \frac{\rho}{2} \|g(x)\|_{2}^{2} \leftarrow \begin{array}{l} \text{Does nothing when} \\ \text{constraint is} \\ \text{s.t.} g(x) = 0 \end{array}$$

Add constraint to objective

Alternating Direction Method of Multipliers (ADMM)

$$\min_{x,z} \quad f(x) + g(z) \\ \text{s.t.} \quad Ax + Bz = c$$

 $\Lambda_{\rho}(x, z; \lambda) = f(x) + g(z) + \lambda^{\top} (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_{2}^{2}$

$$\begin{aligned} x \leftarrow \arg \min_{x} \Lambda_{\rho}(x, z, \lambda) \\ z \leftarrow \arg \min_{z} \Lambda_{\rho}(x, z, \lambda) \\ \lambda \leftarrow \lambda + \rho(Ax + Bz - c) \end{aligned}$$

The Art of ADMM "Splitting"

 $\left\{ \begin{array}{cc} \min_{J} & \sum_{i} \|J_{i}\|_{2} \\ \text{s.t.} & MJ = b \end{array} \right\} \longrightarrow \left\{ \begin{array}{cc} \min_{J,\bar{J}} & \sum_{i} \left(\|J_{i}\|_{2} + \frac{\rho}{2} \|J_{i} - \bar{J}_{i}\|_{2}^{2} \right) \\ \text{s.t.} & M\bar{J} = b \\ J = \bar{J} \end{array} \right\}$ part Takes some practice! Example of "proximal" algorithm.

Solomon et al. "Earth Mover's Distances on Discrete Surfaces." SIGGRAPH 2014.

Want two *easy* subproblems

Frank-Wolfe

To minimize
$$f(x)$$
 s.t. $x \in \mathcal{D}$:
 $s_k \leftarrow \begin{cases} \arg \min_s \ s^\top \nabla f(x_k) \\ \text{s.t.} \ s \in \mathcal{D} \end{cases} \end{cases}$
 $\gamma \leftarrow \frac{2}{k+2}$
 $x_{k+1} \leftarrow x_k + \gamma(s_k - x_k)$

https://en.wikipedia.org/wiki/Frank%E2%80%93Wolfe_algorithm

Linearize objective, preserve constraints