Optimization for machine learning

Nonsmooth optimization

R. Flamary

June 4, 2020

Full course overview

1. **Introduction to numerical optimization**
   1.1 Optimization problem formulation and principles
   1.2 Properties of optimization problems
   1.3 Machine learning as an optimization problem
2. **Constrained Optimization and Standard Optimization problems**
   2.1 Constraints, Lagrangian and KKT
   2.2 Linear Program (LP)
   2.3 Quadratic Program (QP)
   2.4 Other Classical problems (MIP,QCQP,SOCP,SDP)
3. **Smooth Optimization**
   3.1 Gradient descent
   3.2 Newton, quasi-Newton and Limited memory
   3.3 Stochastic Gradient Descent
4. **Non-smooth Optimization**
   4.1 Proximal operator and proximal methods
   4.2 Conditional gradient
5. **Conclusion**
   5.1 Other approaches (Coordinate descent, DC programming)
   5.2 Optimization problem decision tree
   5.3 References an toolboxes

Course overview

Nonsmooth optimization

Optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad F(\mathbf{x}), \tag{1}$$

- F is convex, proper, lower semi-continuous can be non smooth, non continuous.
- Can be constrained optimization with $F(\mathbf{x}) = f(\mathbf{x}) + \chi_\mathcal{C}(\mathbf{x})$.
- General strategy : use the structure of F, find fast iterations.

Optimization strategies

- Subgradient descent: slower than GD ($O(\frac{1}{\sqrt{k}})$), used for training NN.
- Proximal Splitting : divide an conquer strategy, can be accelerated.
- Projected Gradient Descent : special case of proximal splitting.
- Conditional Gradient : Use a linearization of F.

Optimization problem in machine learning

Regularized machine learning

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad f(\mathbf{x}) + g(\mathbf{x}) \tag{2}$$

- f is the data fitting term, g the regularization term.
- Usually f is smooth (K Lipschitz gradient).
- g can be non-smooth for instance Lasso regularization.
- One can use proximal splitting to solve the problem.

Data fiting examples

- Least square:

$$f(\mathbf{x}) = \sum_i (y_i - \mathbf{h}_i^T\mathbf{x})^2$$

- Logistic regression:

$$f(\mathbf{x}) = \sum_i \log(1+\exp(-y_i\mathbf{h}_i^T\mathbf{x}))$$

Regularization examples

- Ridge

$$g(\mathbf{x}) = \frac{\lambda}{2}\sum_k x_k^2$$

- Lasso

$$g(\mathbf{x}) = \lambda\sum_k |x_k|$$

Proximal operator

Definition [Bauschke et al., 2011]

The Proximity (or proximal) operator of a function g is:

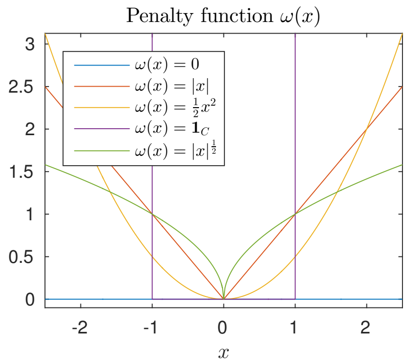
$$\mathbf{prox}_g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$
$$\mathbf{x} \longmapsto \mathbf{prox}_g(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathbb{R}^n} \; g(\mathbf{u}) + \tfrac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2.$$

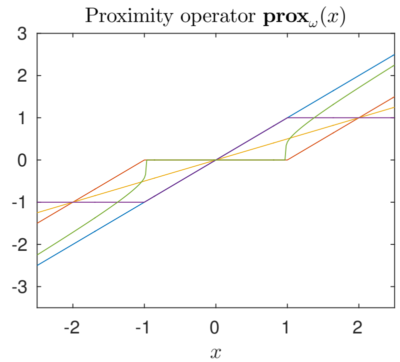
- Returns a vector minimizing g but close to $\mathbf{x}$ in the L2 sens.
- Essential building block of proximal splitting method.

Common proximal operators

$g(\mathbf{x}) = 0$	$\mathbf{prox}_g(\mathbf{x}) = \mathbf{x}$	identity
$g(\mathbf{x}) = \lambda\|\mathbf{x}\|_2^2$	$\mathbf{prox}_g(\mathbf{x}) = \frac{1}{1+\lambda}\mathbf{x}$	scaling
$g(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$	$\mathbf{prox}_g(\mathbf{x}) = \mathrm{sign}(\mathbf{x})\max(0, \|\mathbf{x}\| - \lambda)$	soft shrinkage
$g(\mathbf{x}) = \lambda\|\mathbf{x}\|_{1/2}^{1/2}$	[Xu et al., 2012, Equation 11]	power family
$g(\mathbf{x}) = \chi_C(\mathbf{x})$	$\mathbf{prox}_g(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u}\in C} \frac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2$	orthogonal projection.

Examples of proximal operators





- Proximal operators in 1D.
- Both $|x|$ and $|x|^{\frac{1}{2}}$ promote sparsity (soft thresholds).
- A number of regularization terms are separable:

$$g(\mathbf{x}) = \sum_k w(x_k)$$

Forward Backward Splitting (FBS)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad f(\mathbf{x}) + g(\mathbf{x})$$

FBS algorithm [Combettes and Pesquet, 2011] [Parikh and Boyd, 2014]

1: Initialize $\mathbf{x}^{(0)}$
2: **for** $k = 0, 1, 2, \dots$ **do**
3: $\quad \mathbf{d}^{(k)} \leftarrow -\nabla f(\mathbf{x}^{(k)})$
4: $\quad \mathbf{x}^{(k+1)} \leftarrow \mathbf{prox}_{\rho^{(k)}g}(\mathbf{x}^{(k)} + \rho^{(k)}\mathbf{d}^{(k)})$
5: **end for**

- One gradient step *w.r.t.* f and one proximal step *w.r.t.* g.
- Efficient when the proximal operator is simple to compute (closed form).
- Convergence for a K Lischitz gradient function is $O(\frac{1}{k})$.
- FBS can be generalized to several functions in F [Combettes and Pesquet, 2011]

FBS as Majorization Minimization

- Since f is K gradient Lipschitz F can be bounded by:

$$F(\mathbf{x}) \le f(\mathbf{x}^{(k)}) + \boldsymbol{\nabla} f(\mathbf{x}^{(k)})^t(\mathbf{x}-\mathbf{x}^{(k)}) + \frac{K}{2}\|\mathbf{x}-\mathbf{x}^{(k)}\|^2 + g(\mathbf{x}), \tag{3}$$

- Minimizing the upper bound above is computing $\mathbf{prox}_{\frac{1}{K}g}(\mathbf{x}^{(k)} - \frac{1}{K}\nabla f(\mathbf{x}^{(k)}))$

FBS Acceleration

FBS with Nesterov acceleration [Beck and Teboulle, 2009]

Initialize $\mathbf{y}^{(1)} = \mathbf{x}^{(0)}, t^{(1)} = 1$
for $k = 1, 2, \dots$ **do**
  $\mathbf{x}^{(k)} \leftarrow \mathbf{prox}_{\rho^{(k)} g}(\mathbf{y}^{(k)} - \rho^{(k)} \nabla f(\mathbf{y}^{(k)}))$
  $t^{(k+1)} \leftarrow \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$
  $\mathbf{y}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \frac{t^{(k)}-1}{t^{(k+1)}}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$
end for

- Use a similar momentum to accelerated gradient.
- Convergence in value is $O(\frac{1}{k^2})$.
- The function might not decrease at each iteration due to the momentum.

Adaptive step [Goldstein et al., 2014]

Compute the step $\rho^{(k)}$ with the Barzilai-Borwein rule [Barzilai and Borwein, 1988]:

$$\rho_s = \frac{\Delta\mathbf{x}^T\Delta\mathbf{x}}{\Delta\mathbf{x}^T\Delta\mathbf{g}} \quad \text{and} \quad \rho_m = \frac{\Delta\mathbf{x}^T\Delta\mathbf{g}}{\Delta\mathbf{g}^T\Delta\mathbf{g}}$$

With $\Delta\mathbf{x} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$ and $\Delta\mathbf{g} = \nabla f(\mathbf{y}^{(k)})) - \nabla f(\mathbf{y}^{(k-1)}))$. This corresponds to estimate locally the Hessian matrix as $\sigma\mathbf{I}$.

Exercise 1: solving the Lasso with FBS

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 + \lambda \sum_k |x_k|$$

1. Express the smooth function f and non-smooth functions g for the problem above

   $f(\mathbf{x}) =$ $\qquad\qquad$ $g(\mathbf{x}) =$

2. Compute the gradient $\nabla f(\mathbf{x})$ and express the proximal of g.

   $\nabla f(\mathbf{x}) =$ $\qquad\qquad$ $\mathbf{prox}_g(\mathbf{x}) =$

3. Express the FBS algorithm in Python/Numpy for solving the lasso with a fixed step `rho` :

Alternating Direction Method of Multipliers (ADMM)

Optimization problem and augmented Lagrangian

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{z}\in\mathbb{R}^m} \quad f(\mathbf{x}) + g(\mathbf{z}) \qquad (4)$$
$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}$$

The augmented Lagrangian of the problem is expressed as:

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2 \qquad (5)$$

ADMM Algorithm [Boyd et al., 2011]

1: Initialize $\mathbf{x}^{(0)}, \mathbf{z}^{(0)}, \mathbf{y}^{(0)}, \rho > 0$
2: **for** $k = 1, 2, \dots$ **do**
3:   $\mathbf{x}^{(k+1)} \leftarrow \arg\min_{\mathbf{x}} \ L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)})$
4:   $\mathbf{z}^{(k+1)} \leftarrow \arg\min_{\mathbf{z}} \ L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \mathbf{y}^{(k)})$
5:   $\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c})$
6: **end for**

- Updates 3 and 4 can often be expressed as proximal updates.
- When f or g is separable, the updates can be done in parallel.

Primal-Dual Algorithms

Douglas-Rachford Splitting

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x}) + g(\mathbf{x})$$

Initialize $\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \rho > 0$
for $k = 1, 2, \dots$ **do**
  $\mathbf{x}^{(k+1)} \leftarrow \mathbf{prox}_f(\mathbf{y}^{(k)})$
  $\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} + \mathbf{prox}_g(2\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)})) - \mathbf{x}^{(k+1)}$
end for

Chambole-Pock [Chambolle and Pock, 2011]

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{A}\mathbf{x}) + g(\mathbf{x})$$

Both f and g are convex, their proximal can be computed efficiently.

Vu-Conda Algorithm [Vũ, 2013, Condat, 2014]

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{A}\mathbf{x})$$

- f convex with K Lipschitz gradients.
- g and h are convex and have "simple" proximity operators.

Example: Total Variation denoising

$$\min_{\mathbf{X}\in\mathbb{R}_+^{d\times d}} \quad \|\mathbf{Y}-\mathbf{X}\|_F^2 + \lambda\left(\sum_{i=1,j=1}^{d,d-1}|X_{i,j}-X_{i,j+1}| + \sum_{i=1,j=1}^{d-1,d}|X_{i,j}-X_{i+1,j}|\right)$$

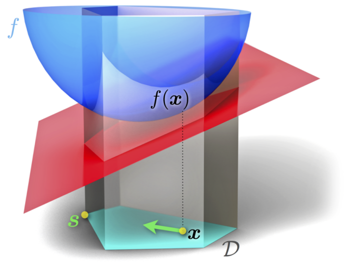
- Image $\mathbf{Y}$ is supposed to be noisy. We want to recover a clean $\mathbf{X}$ that has piecewise constant parts.
- The regularization term measure the total variation (2D gradients) of the image horizontally and vertically.
- The optimization problem can be expressed as:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{A}\mathbf{x})$$

- It can be solved using ADMM, Chambole-Pock or Vu-Conda.

Conditional Gradient method

$$\min_{\mathbf{x}\in\mathcal{C}} \quad F(\mathbf{x})$$



Algorithm

1: Initialize $\mathbf{x}^{(0)} \in \mathcal{C}$
2: **for** $k = 0, 1, 2, \dots$ **do**
3: $\quad \mathbf{s}^{(k)} \leftarrow \arg\min_{\mathbf{s}} \mathbf{s}^T\nabla F(\mathbf{x}^{(k)})$, s.t. $\mathbf{s}\in\mathcal{C}$
4: $\quad \rho^{(k)} \leftarrow$ compute step size $\rho\in[0,1]$
5: $\quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho^{(k)}(\mathbf{s}^{(k)} - \mathbf{x}^{(k)})$
6: **end for**

- Proposed in [Frank and Wolfe, 1956] to solve Quadratic Programs.
- Also known as the Frank-Wolfe algorithm.
- When $\mathcal{C}$ correspond to linear constraints each iteration is a LP.
- The step $\rho^{(k)}$ can be either decreasing or estimated via linesearch:

$$\rho^{(k)} = \frac{2}{k+2} \qquad \text{or} \qquad \rho^{(k)} = \underset{\rho\in[0,1]}{\text{argmin}}\, F(\mathbf{x}^{(k)} + \rho(\mathbf{s}^{(k)} - \mathbf{x}^{(k)}))$$

- Reintroduced in ML recently [Jaggi, 2013].

Image courtesy of Martin Jaggi

CG convergence and certificate

Lower bound on the optimal value

- Since F is convex one has:

$$\begin{aligned} F(\mathbf{x}^\star) &\geq F(\mathbf{x}) + (\mathbf{x}^\star - \mathbf{x})^T\nabla F(\mathbf{x}) \\ &\geq \min_{\mathbf{y}\in\mathcal{C}} \left\{F(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^T\nabla F(\mathbf{x})\right\} \\ &= F(\mathbf{x}) + \mathbf{x}^T\nabla F(\mathbf{x}) + \min_{\mathbf{y}\in\mathcal{C}} \mathbf{y}^T\nabla F(\mathbf{x}) \end{aligned}$$

- This lower bound can be computed at each iteration as :

$$F(\mathbf{x}^{(k)}) + (\mathbf{s}^{(k)} - \mathbf{x}^{(k)})^T\nabla F(\mathbf{x}^{(k)})$$

Certificate and convergence

- From the bound above we have the following certificate:

$$l_k \leq F(\mathbf{x}^\star) \leq F(\mathbf{x}^{(k)}) \quad \text{with} \quad l_k = \max(l_k{-}1, F(\mathbf{x}^{(k)}) + (\mathbf{s}^{(k)} - \mathbf{x}^{(k)})^T\nabla F(\mathbf{x}^{(k)}))$$

- Converges to the optimal value in $O(\frac{1}{k})$ [Jaggi, 2013] .
- Also converges when F is smooth and non-convex [Lacoste-Julien, 2016].

Exercise 2: CG for Lasso with constraints

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} \quad & \frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \leq \tau \end{aligned}$$

1. Find the solution for the following optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad \mathbf{x}^T\mathbf{g}, \qquad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq \tau$$

$$x_i^\star =$$

2. Code in Python/Numpy a solver using the decreasing step.

Conclusion

Proximal methods [Parikh and Boyd, 2014]

- General strategy of proximal splitting: divide and conquer the objective function.
- Search for a stationary point, avoid subgradients.
- FBS for simple problems, ADMM or other Primal/Dual approaches for more complex splitting.
- Very efficient when proximal have a closed form.
- For sparse optimization, intermediate iterates are sparse.
- Works also for non-convex problems [Attouch et al., 2010].

Conditional Gradient

- Solve iteratively linearization of the function under constraints.
- Very efficient if the linearized problem has a closed form.
- Can be extended with linearization of only one part of the function [Bredies et al., 2009]

References I

[Attouch et al., 2010] Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. (2010).
Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-Lojasiewicz inequality.
Mathematics of Operations Research, 35(2):438–457.

[Barzilai and Borwein, 1988] Barzilai, J. and Borwein, J. M. (1988).
Two-point step size gradient methods.
IMA Journal of Numerical Analysis, 8(1):141–148.

[Bauschke et al., 2011] Bauschke, H. H., Combettes, P. L., et al. (2011).
Convex analysis and monotone operator theory in Hilbert spaces, volume 408.
Springer.

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009).
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM journal on imaging sciences, 2(1):183–202.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011).
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends® in Machine learning, 3(1):1–122.

References II

[Bredies et al., 2009] Bredies, K., Lorenz, D. A., and Maass, P. (2009).
A generalized conditional gradient method and its connection to an iterative shrinkage method.
Computational Optimization and Applications, 42(2):173–193.

[Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011).
A first-order primal-dual algorithm for convex problems with applications to imaging.
Journal of mathematical imaging and vision, 40(1):120–145.

[Combettes and Pesquet, 2011] Combettes, P. L. and Pesquet, J.-C. (2011).
Proximal splitting methods in signal processing.
In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer.

[Condat, 2014] Condat, L. (2014).
A generic proximal algorithm for convex optimization—application to total variation minimization.
IEEE Signal Processing Letters, 21(8):985–989.

[Frank and Wolfe, 1956] Frank, M. and Wolfe, P. (1956).
An algorithm for quadratic programming.
Naval research logistics quarterly, 3(1-2):95–110.

References III

[Goldstein et al., 2014] Goldstein, T., Studer, C., and Baraniuk, R. (2014).
A field guide to forward-backward splitting with a fasta implementation.
arXiv preprint arXiv:1411.3406.

[Jaggi, 2013] Jaggi, M. (2013).
Revisiting frank-wolfe: Projection-free sparse convex optimization.
In *Proceedings of the 30th international conference on machine learning*, number CONF, pages 427–435.

[Lacoste-Julien, 2016] Lacoste-Julien, S. (2016).
Convergence rate of frank-wolfe for non-convex objectives.
arXiv preprint arXiv:1607.00345.

[Parikh and Boyd, 2014] Parikh, N. and Boyd, S. P. (2014).
Proximal algorithms.
Foundations and Trends in optimization, 1(3):127–239.

[Vũ, 2013] Vũ, B. C. (2013).
A splitting algorithm for dual monotone inclusions involving cocoercive operators.
Advances in Computational Mathematics, 38(3):667–681.

[Xu et al., 2012] Xu, Z., Chang, X., Xu, F., and Zhang, H. (2012).
$L_{1/2}$ regularization: a thresholding representation theory and a fast solver.
Neural Networks and Learning Systems, IEEE Transactions on, 23(7):1013–1027.