Algorithms inspired by Lagrange duality

$$\boxed{\begin{aligned} &\min \; f(x) \\ &\text{s.t } Ax = b \end{aligned}}$$

KKT: $L(x,\nu) = f(x) + \nu^T(Ax-b)$

$\nabla_x L(x^*,\nu^*) = \nabla f(x^*) + A^T\nu^* = 0$ $\{*\}$

If $Ax \le b$ then additionally: $\nu^* \ge 0$

& :. $\nu^{(k+1)} = [\nu^{(k)} + t^{(k)}(Ax^{(k+1)} - b)]_+$

componentwise thresholding

Note: $\partial_\nu L(x,\nu) = Ax - b$

Dual ascent

$$\boxed{\begin{aligned} x^{(k+1)} &= \underset{x}{\operatorname{argmin}} \; L(x,\nu^{(k)}) = \underset{x}{\operatorname{argmin}} \; f(x) + (\nu^{(k)})^T Ax \\ \nu^{(k+1)} &= \nu^{(k)} + t^{(k)} \partial_\nu L(x^{(k+1)},\nu^{(k)}) = \nu^{(k)} + t^{(k)}(Ax^{(k+1)} - b) \end{aligned}}$$

If strong duality holds & $\nu^{(k)} \to \nu^{(*)}$ then

$x^{(*)} = \underset{x}{\operatorname{argmin}} \; L(x,\nu^*)$ provided minimizer is unique

$O(1/k)$ convergence if f is Lipschitz continuous with const d & provided $t^{(k)} \le d \;\; \forall k$.

Dual decomposition

$$\boxed{f(x) = \sum_{i=1}^{b} f_i(x_i)}$$

$x = [x_1, x_2 \ldots x_b]$ .. b blocks of variables

Simplification of dual ascent (useful for parallelization)

More robust: Method of Augmented Lagrangian

Apply Dual descent to

$$\begin{aligned} \min_x \;\; & f(x) + \frac{\rho}{2}\|Ax-b\|^2 \\ \text{s.t} \;\; & Ax = b \end{aligned}$$

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}} \; \underbrace{f(x) + \frac{\rho}{2}\|Ax-b\|^2 + (\nu^{(k)})^T(Ax-b)}_{L_\rho(x,\nu)}$$

$$\nu^{(k+1)} = \nu^k + \rho(Ax^{(k)} - b)$$

$$x^{(k+1)} \doteq \underset{x}{\operatorname{argmin}}\ f(x) + \nu^{(k)T} A x$$

$$= \underset{x_1 \ldots x_b}{\operatorname{argmin}} \sum_i f_i(x_i) + \nu^{(k)T}\left(\sum_i A_i x_i\right)$$

$(A = [A_1\ A_2 \ldots A_b])$

Thus:

$$\boxed{x_i^{(k+1)} = \underset{x_i}{\operatorname{argmin}}\ f_i(x_i) + \nu^{(k)T} A_i x_i}$$

for each block, possibly in parallel after broadcasting $\nu^{(k)}$

$$\boxed{\nu^{(k+1)} = \nu^{(k)} + t^{(k)}\left(A x^{(k+1)} - b\right)}$$

After gathering $x_i^{(k+1)}$ from all nodes & updating $\nu^{(k+1)}$ for again broadcasting

$t^{(k)} = \rho$ motivated by observation that necessary condition for $x^{(k+1)}$ is

$$0 \in \partial f(x^{(k+1)}) + A^T\big(\underbrace{\nu^{(k)} + \rho\,(A x^{(k)} - b)}_{y^{(k+1)}}\big)$$

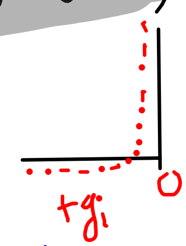
ie $0 \in \partial f(x^{(k+1)}) + A^T y^{(k+1)}$

Comparing with $\{*\}$ expression for $y^{(k+1)}$ can be update rule for $\nu^{(k+1)}$!

Dual descent on Augmented Lagrangian has better conver--gence rate

Interior Point Methods (IPM) (Barrier method... like augmented Lagrangian)

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}}\ f(x) - \frac{1}{t^{(k)}} \sum \log(-g_i(x^k))$$



ADMM (Alternating direction Method of Multipliers)

$$f(x) = \sum f_i(x_i),\quad L_\rho(x_1 \ldots x_b, \nu) = \sum f_i(x_i) + \frac{\rho}{2}\|Ax - b\|^2 + \nu^T(Ax - b)$$

$$x_1^{(k+1)} = \underset{x_1}{\operatorname{argmin}}\ L_\rho(x_1, x_2^k, \ldots \nu^k),\quad x_2^{(k+1)} = \underset{x_2}{\operatorname{argmin}}\ L_\rho(x_1^{(k+1)}, x_2, x_3^{(k)} \ldots \nu^k)$$

$$x_3^{(k+1)} = \underset{x_3}{\operatorname{argmin}}\ L_\rho(x_1^{(k+1)}, x_2^{(k+1)}, x_3, x_4^{(k)} \ldots \nu^{(k)}) \ldots x_b^{(k+1)} = \underset{x_b}{\operatorname{argmin}}\ L_\rho(x_1^{(k+1)}, \ldots x_{b-1}^{(k+1)}, x_b, \nu^{(k)})$$

$$\nu^{(k+1)} = \nu^k + \rho\left(A_1 x_1^{(k+1)} + A_2 x_2^{(k+1)} \cdots + A_b x_b^{(k+1)} - b\right)$$

Other methods (First order: Require only upto $\nabla f(x)$)

① Mirror descent

Recap: Gradient descent (subgradient descent)

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}}\ f(x^k) + \partial^T f(x^k)(x - x^k) + \frac{t}{2}\|x - x^k\|^2$$

Mirror descent:

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}}\ f(x^k) + \partial^T f(x^k)(x - x^k) + \Delta_g(x, x^k)$$

(Bregman Divergence)

② Accelerated gradient descent methods

$$x^{(k+1)} = F(x^{(k)}, x^{(k-1)}, \ldots)$$

- First iteration is just usual proximal gradient descent update
  - After that, updates carry some "momentum" from previous iterations
  - Example: Conjugate gradient methods: Subtracts previous descent direction from the current gradient to give a current descent direction that is steep but is nearly orthogonal to previous descent directions. Pages 317 to 324 of http://www.cse.iitb.ac.in/~cs709/notes/BasicsOfConvexOptimization.pdf. In particular, see Figure 4.55
  - LBFGS | BFGS (see notes ...)

③ ACTIVE SET METHOD

http://www.cse.iitb.ac.in/~cs709/notes/quadraticOpt-PrimalActiveSet.pdf

④ CUTTING PLANE METHOD

http://www.cse.iitb.ac.in/~cs709/notes/kellysCuttingPlaneAlgo.pdf

5) Second order methods (Newton's method)

http://www.cse.iitb.ac.in/~cs709/notes/BasicsOfConvexOptimization.pdf (pg 305)