# Other types of methods for large-scale optimization

*Computational Methods in Optimization*
*CS 520, Purdue*

David F. Gleich
Purdue University

# ALTERNATING OPTIMIZATION

# Non-negative matrix factorization

$$\begin{array}{ll} \text{minimize} & \|\mathbf{A} - \mathbf{X}\mathbf{Y}^T\| \\ \text{subject to} & \mathbf{X} \in \mathbb{R}^{m \times k} \geq 0, \mathbf{Y} \in \mathbb{R}^{n \times k} \\ & \mathbf{X} \geq 0, \mathbf{Y} \geq 0 \end{array}$$

Fix X, solve for Y

Fix Y, solve for X

…

Does it converge?

# Block coordinate descent

*Gauss-Seidel*

*Alternating direction*

Lots of activity among machine learning, compressed sensing, sparse 1-norm folks, too.

# Bertsekas, Nonlinear programming

Suppose f is continuous, differentiable

$f = f(x_1, \ldots, x_N)$ where $x_i$ is in a convex domain. "Think of each $x_i$ as a block of variables."

If

$$\underset{\mathbf{y} \in X_i}{\text{minimize}}\, f(\mathbf{x}_1, \ldots, \mathbf{y}, \ldots, \mathbf{x}_N)$$

is uniquely attained, then the sequence of subproblems converges to a stationary point.

# Suppose there are just two blocks

[Grippo & Sciandrone]

Then we don’t need a unique minimizer any more and we can treat more general convex problems.

# More general setting

*Alternating Direction Method of Multipliers*

- More general problem theory,
- Take your problem and break it up into “solvable” pieces and then put a Lagrange multiplier on the equality

e.g. min f(x) s.t. Ax=b -> min f(x) s.t. Ay = b, x = y

e.g. min f(x) + ||x|| -> min f(x) + ||y|| s.t. x = y

solve for x given y given Lagrangian mults on x=y,
solve for y given _new_ x given Lagrangian mults on x=y,
update Lagrangian mults.

# An example of ADMM

## Overlapping, non exhaustive clustering

$$
\begin{array}{ll}
\underset{\boldsymbol{Y},\mathbf{f},\mathbf{g},\mathbf{s},r}{\text{minimize}} & \mathbf{f}^T\mathbf{d} - \operatorname{trace}(\boldsymbol{Y}^T\boldsymbol{K}\boldsymbol{Y}) \\
\text{subject to} & k = \operatorname{trace}(\boldsymbol{Y}^T\boldsymbol{W}^{-1}\boldsymbol{Y}) \\
& 0 = \boldsymbol{Y}\boldsymbol{Y}^T\mathbf{e} - \boldsymbol{W}\mathbf{f} \\
& 0 = \mathbf{e}^T\mathbf{f} - (1+\alpha)n \\
& 0 = \mathbf{f} - \mathbf{g} - \mathbf{s} \\
& 0 = \mathbf{e}^T\mathbf{g} - (1-\beta)n - r \\
& Y_{i,j} \geq 0, \mathbf{s} \geq 0, r \geq 0 \\
& 0 \leq \mathbf{f} \leq k\mathbf{e}, 0 \leq \mathbf{g} \leq 1
\end{array}
$$

$$
\begin{aligned}
\boldsymbol{Y}^{k+1} &= \underset{\boldsymbol{Y}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\boldsymbol{Y}, \mathbf{f}^k, \mathbf{g}^k, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \gamma^k, \sigma) \quad \text{Apply bounds on Y} \\
\mathbf{f}^{k+1} &= \underset{\mathbf{f}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\mathbf{Y}^{k+1}, \mathbf{f}, \mathbf{g}^k, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \gamma^k, \sigma) \quad \text{Apply bounds on f} \\
\mathbf{g}^{k+1} &= \underset{\mathbf{g}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\mathbf{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}, \mathbf{s}^k, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \gamma^k, \sigma) \quad \text{Apply bounds on g} \\
\mathbf{s}^{k+1} &= \underset{\mathbf{s}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\mathbf{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}^{k+1}, \mathbf{s}, r^k; \\
&\qquad \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \gamma^k, \sigma) \quad \text{Apply bounds on s} \\
r^{k+1} &= \underset{r}{\operatorname{argmin}} \mathcal{L}_{\mathcal{A}}(\mathbf{Y}^{k+1}, \mathbf{f}^{k+1}, \mathbf{g}^{k+1}, \mathbf{s}^{k+1}, r;
\end{aligned}
$$

Hou, Whang, Gleich, Dhillon. Fast Multiplier Methods for Non-exhaustive Overlapping Clustering

# STOCHASTIC GRADIENT DESCENT

# SGD

Given $f(\mathbf{x}) = \sum_{i=1}^{L} f_i(\mathbf{x})$.

Note that $\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{L} \nabla f_i(\mathbf{x})$

Consider $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla f_{i \sim U}(\mathbf{x})$

Here, $\nabla f_{i \sim U}(\mathbf{x})$ is just a random term in the gradient ("i drawn from uniform U")

# Stochastic Gradient Descent

$$\text{minimize} \quad \|\mathbf{Ax} - \mathbf{b}\|^2$$

$$\text{minimize} \quad \sum_i \left( \sum_j A_{ij} x_j - b_i \right)^2$$

$$\text{minimize} \quad \textstyle\sum_i \ell_i(\mathbf{x})$$

$$\ell_i(\mathbf{x}) = \left( \sum_j A_{ij} x_j - b_i \right)^2$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \mathbf{g}_{\ell_i}(\mathbf{x}^{(k)})$$

Repeatedly draw $i$ at random.

$$= \mathbf{x}^{(k)} - \alpha 2 \left(\sum_j A_{ij} x_j - b_i\right) \begin{bmatrix} A_{i,1} \\ \vdots \\ A_{i,n} \end{bmatrix}$$

# Examples that aren't separable harder to think of how SGD would work

- max time
  s.t. the equations of motion (e.g. raptor)
- min cost
  s.t. the object is buildable
- min fuel
  s.t. we get to mars

Reformulate to min / max expected quantity over trajectory