Lecture 18 - BCFW

Today

* FW for SUMstruct & FCFW
* BCFW

Continue FW for SUMstruct

FW on dual is equivalent to batch subgradient update on primal with step-size rule:
\[ x(t) \rightarrow \mathbf{w}^{(t)} = \mathbf{A} x(t) \]
\[ y^{(FW)} = \beta \] (subgradient)

Recall: subgradient method converges
\[ O\left(\frac{1}{\epsilon}\right) \]
when step-size \( \beta \) s.t. \( O\left(\frac{1}{\epsilon}\right) \)

FW method with
principal step-size
\[ \gamma_t = \frac{2}{t+2} \]
gives
\[ d(\alpha^*) - d(\alpha^{(t)}) \leq \frac{2C_\epsilon}{t+2} \]

* FW gap:

\[ \mathbf{p}(\mathbf{w}^{(t)}) - d(\alpha^{(t)}) \]

here FW gap = Lagrangian gap on \((\mathbf{w}^{(t)}, \alpha^{(t)})\)

[Note: FW gap is not always lag. gap]

Also, CRF objective on (dual)

Recall that
\[ \min_{\delta \in \mathcal{T}} \delta \mathbf{g}_N \leq 3 \cdot \frac{2C_\epsilon}{t+2} \]

\[ \Rightarrow \text{guarantees on } \mathbf{p}(\mathbf{w}(\mathbf{a})) - \mathbf{p}(\mathbf{w}^*) + d(\mathbf{a}^*) - d(\mathbf{a}(\mathbf{t})) \]

(when \( \mathbf{g}(\cdot) \) is convex [this is case when \( f \) is...
\[ p(w|a^{(t)}) = \frac{1}{n} \max_{i=1}^{n} f_i(x) \]

\[ \rho^* = d^* \]

\[ \text{FW gap}(a|e^{(t)}) = \log \text{gap} \]

**FCFW “fully corrective FW” variant**

**Algorithm:** Re-optimizing convex hull \( \text{conv-hull}(\sum_{i=1}^{n} e_i^{(t)}) \)

→ think of it as doing “fancy line search”

on the “corrected polytope”

**Note:** Could use AFW to do correction step approx.

(see “bouncy FW”)

[ Special case: min norm point alg, MIR ]

→ sequence of affine projectors

→ line search to approximates

the correction step

→ state of the art alg. for submodular opt.

**Turns out that (batch) FCFW on dual weakly

is equivalent to the constraint generation/cutting plane alg

on the 1-slack formulation (proven)**

why? every \(s^{(t)}\) corresponds to \( \lambda_i^{(t)} \)

\[ \alpha \in \sum_{u \in U} s^{(t)} \]

Solving O(n) primal for cutting plane

correction step and for FCFW

<table>
<thead>
<tr>
<th>Dual problem</th>
<th>Dual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O(n) ) oracle calls quit within</td>
<td>batch subgradient</td>
</tr>
<tr>
<td>( \beta_t = \frac{\lambda}{\alpha} )</td>
<td>1-slack cutting plane</td>
</tr>
</tbody>
</table>

\[ W = A \alpha = \sum_{u \in U} s^{(t)} \]

Complexity of alg. less in \#oracle calls

Augmented dual

\[ O(n^2 \log^k \beta) \rightarrow n \geq 1? \]

\[ O(n \log \beta) \]

Open problem related to...
Block-coordinate optimization

There are several popular optimization methods in use, such as gradient descent, stochastic gradient descent, and their variants. For large-scale problems, block-coordinate methods are often used. These methods update only a subset of the variables at each iteration, which can be more efficient than updating all variables.

### Stochastic Subgradient Method

- **Input:** Function $f: \mathbb{R}^n \to \mathbb{R}$, initial point $x_0$, step size $\alpha > 0$.
- **Procedure:**
  - For $t = 1, 2, \ldots$
  - Sample a subset $A_t \subseteq \{1, 2, \ldots, n\}$ of size $m$.
  - Compute a subgradient $g_t$ of $f$ at $x_t$.
  - Update $x_{t+1}$ using $x_{t+1} = x_t - \alpha g_t$.

### Block-Coordinate FW (BCFW)

- **Setup:** $f: \mathbb{R}^n \to \mathbb{R}$, $x_0 \in \mathbb{R}^n$, $\{A_t\}$.
- **Algorithm:**
  - For $t = 1, 2, \ldots$
  - Sample a subset $A_t \subseteq \{1, 2, \ldots, n\}$ of size $m$.
  - For $i \in A_t$
    - $x_t^{(i)} = x_t^{(i)} - \alpha_t g_t^{(i)}$
    - $x_t^{(i)} = x_t^{(i)} + \alpha_t s_t^{(i)}$
  - $x_{t+1} = \min_{x \in \mathbb{R}^n} f(x) + \sum_{i \in A_t} \langle g_t^{(i)}, x_t^{(i)} - x \rangle$

### Notes

- The block-coordinate FW method is particularly useful for problems where the function $f$ is separable or can be efficiently evaluated block-wise.

- The method can be accelerated using techniques such as the Nesterov acceleration, which reduces the convergence rate of the algorithm.

- For strongly convex functions, the convergence rate of the block-coordinate FW method can be improved by choosing the step sizes appropriately.

- The choice of block size $m$ and the subset selection strategy can significantly impact the performance of the algorithm.

---

*IFT6132 Page 3*
$\Phi$ an important property: $\text{FW-gap} = \max_i \langle -Df_i(x), s-x \rangle \leq \max_i \langle -Df_i(x), s-x \rangle \leq g_i(x) \leq g_i(x) \leq \text{back FW gap}$

As before, you can show that $g_i(x) \geq f(x) - \min_i f_i(x)$

[Osokin et al., ICML 2015]

Convergence of BCFW

$C_E^{(i)} \leq L \cdot \text{diam}(X_i)^2$

$C_E \leq L \cdot \sum_{i=1}^{n} C_i$

$\frac{E f(x^{(t)}) - f^*}{\|x^t - x^{(t)}\|_2} \leq \frac{2n}{t+2n}$

if you use line search $\frac{2n C_{E_{\text{line}}}}{t+2n}$ for $t \geq 0$

time to ensure that $\|x^t - x^{(t)}\|_2 \leq \sqrt{n \log 2 |f(x^{(t)})|}$

Batch FW

$C_f \leq 2C_f + \frac{2}{t+2}$

BCFW

$C_f \leq \frac{2n C_{E_{\text{line}}}}{t+2n}$

one can show that $C_f \lesssim f_{\text{quad}}$ for quadratic functions

$C_E$ vs. $n C_E$ (complexity)

BCFW is $n$ times cheaper than Batch FW (unlike FW)

BCFW is "never" slower than Batch FW

but for unstructured $\text{quad}$, $n C_E$ is actually $n$ times smaller than $C_f$

Extensions to BCFW:

- non-uniform sampling $\mathbb{E}_{i} \left\{ C_i \right\}$

ICML 2016
- Non-uniform sampling e.g. \[ G(x) \]

- Using away-step, etc. to get linear convergence