Summary

Block-Coordinate Frank-Wolfe (BC-FW) [1] is a popular algorithm for constrained optimization over block-separable domains. Examples: dual of Structured SVM, multiple-sequence alignment, etc.

Key insight: Frank-Wolfe (FW) block gaps indicate suboptimality on blocks and we use them to design adaptive algorithms.

Contributions:
1. Adaptive sampling of blocks using FW gaps
2. Caching the oracle calls with gap-based criterion
3. Pairwise and away steps in BC setting
4. Regularization path for Structured SVM

Motivation

Problem: The oracle is often bottleneck
Solution: cache the outputs of the oracle and reuse them

Cache oracle: \( g(y) := \max_{y \in Y} \frac{1}{\beta} \sum_{i=1}^n \ell_i(y) - u(y) \)

Possible improvement: \( g(y) := \max_{y \in Y} \frac{1}{\beta} \sum_{i=1}^n \ell_i(y) - u(y) \alpha \)

Overheads for maintaining cache
Safety rate of \( O(1/\varepsilon) \)
Open question: add cost in analysis

Adaptive criterion

When to use the cache?

\[ \tilde{g}(k) = \max \{ F_k, g(k) \} \]

Experimental results

Extensive benchmark
- Comparison of 8 methods
- Tested on 4 structured prediction datasets:
  OCR / CoNLL / HorseSeg / LSP  
  NLP / Vision

Conclusions
- 1. Gap sampling always helps
- 2. Caching really helps if slow oracle (cost of our naïve implementation)
- 3. BC-A-FW helps for high accuracy and stronger when more strongly convex

References