Limited Memory Kelley's Method Converges for Composite Convex and Submodular Objectives

Madeleine Udell

Operations Research and Information Engineering
Cornell University

Based on joint work with
Song Zhou (Cornell) and Swati Gupta (Georgia Tech)

Rutgers, 11/9/2018

My old 2013 Macbook Pro: 16GB Ram

Gonna buy a new model with more RAM. . .

View gallery

Which processor is right for you?

2.3GHz quad-core 8th-generation Intel Core i5 processor, Turbo Boost up to 3.8GHz - $300.00

2.7GHz quad-core 8th-generation Intel Core i7 processor, Turbo Boost up to 4.5GHz

Memory

How much memory is right for you?

8GB 2133MHz LPDDR3 memory - $200.00

16GB 2133MHz LPDDR3 memory

Gonna buy a new model with more RAM...

View gallery

Which processor is right for you?

2.3GHz quad-core 8th-generation Intel Core i5 processor, Turbo Boost up to 3.8GHz	- $300.00
2.7GHz quad-core 8th-generation Intel Core i7 processor, Turbo Boost up to 4.5GHz	

Memory

How much memory is right for you?

8GB 2133MHz LPDDR3 memory	- $200.00
16GB 2133MHz LPDDR3 memory	

nope! RAM in 13in Macbook Pro $\leq$ 16 GB.

Ok, so RAM isn't smaller. Is it cheaper?

13-inch | 15-inch

Touch Bar and Touch ID 2.2GHz 6-Core Processor 256GB Storage

2.2GHz 6-core 8th-generation Intel Core i7 processor

Turbo Boost up to 4.1GHz

Radeon Pro 555X with 4GB of GDDR5 memory

16GB 2400MHz DDR4 memory

256GB SSD storage[1]

Retina display with True Tone

Touch Bar and Touch ID

Four Thunderbolt 3 ports

$2,399.00

Touch Bar and Touch ID 2.6GHz 6-Core Processor 512GB Storage

2.6GHz 6-core 8th-generation Intel Core i7 processor

Turbo Boost up to 4.3GHz

Radeon Pro 560X with 4GB of GDDR5 memory

16GB 2400MHz DDR4 memory

512GB SSD storage[1]

Retina display with True Tone

Touch Bar and Touch ID

Four Thunderbolt 3 ports

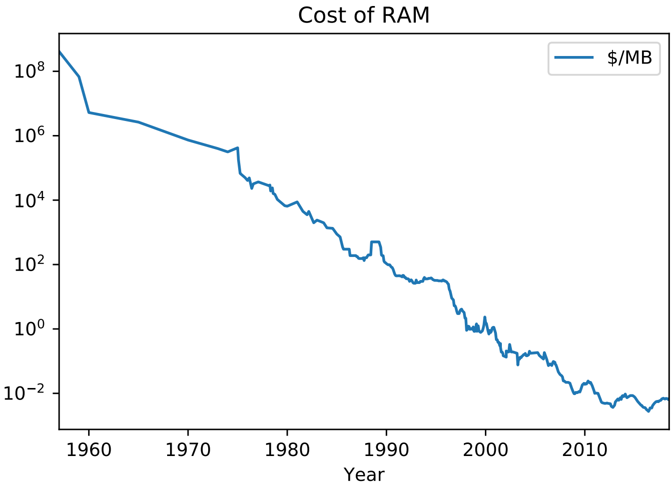
$2,799.00

Ok, so RAM isn't smaller. Is it cheaper?

13-inch	15-inch
Touch Bar and Touch ID 2.2GHz 6-Core Processor 256GB Storage	**Touch Bar and Touch ID 2.6GHz 6-Core Processor 512GB Storage**
2.2GHz 6-core 8th-generation Intel Core i7 processor	2.6GHz 6-core 8th-generation Intel Core i7 processor
Turbo Boost up to 4.1GHz	Turbo Boost up to 4.3GHz
Radeon Pro 555X with 4GB of GDDR5 memory	Radeon Pro 560X with 4GB of GDDR5 memory
16GB 2400MHz DDR4 memory	16GB 2400MHz DDR4 memory
256GB SSD storage[1]	512GB SSD storage[1]
Retina display with True Tone	Retina display with True Tone
Touch Bar and Touch ID	Touch Bar and Touch ID
Four Thunderbolt 3 ports	Four Thunderbolt 3 ports
$2,399.00	**$2,799.00**

Nope!

RIP Moore's Law for RAM (circa 2013)



Why low memory convex optimization?

low memory

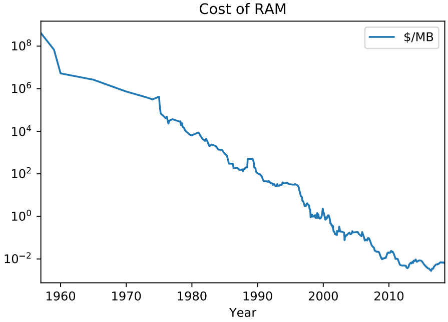
- Moore's law is running out
- low memory algorithms are often fast

convex optimization

- robust convergence
- elegant analysis

Memory plateau and conditional gradient method

- (Frank & Wolfe 1956) An algorithm for quadratic programming
- (Levitin & Poljak 1966) "Conditional gradient method"
- (Clarkson 2010) Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm
- (Jaggi 2013) Revisiting Frank-Wolfe: projection-free sparse convex optimization



Example: smooth minimization over ℓ_1 ball

for $g : \mathbb{R}^n \to \mathbb{R}$ smooth, $\alpha \in \mathbb{R}$, find iterative method to solve

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & \|w\|_1 \leq \alpha \end{array}$$

what kinds of subproblems are easy?

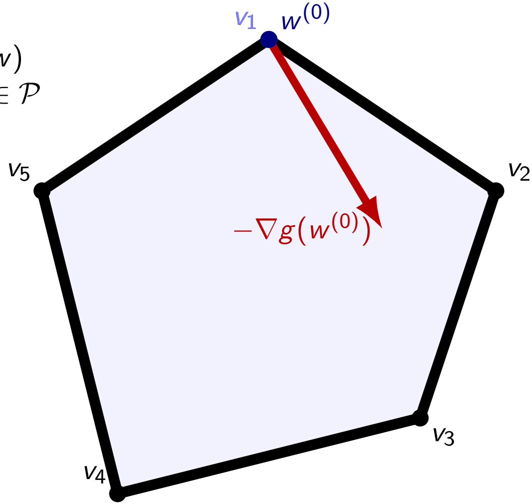
- projection is complicated (Duchi et al. 2008)
- linear optimization is easy:

$$\alpha e_i = \begin{array}{ll} \text{argmin} & x^\top w \\ \text{subject to} & \|w\|_1 \leq \alpha \end{array}$$

where $i = \mathrm{indmax}(w)$

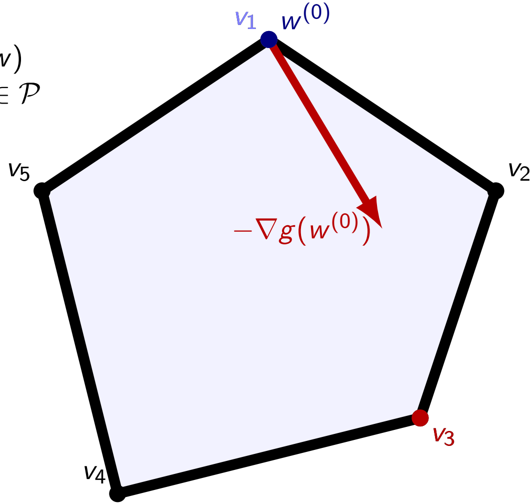
Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



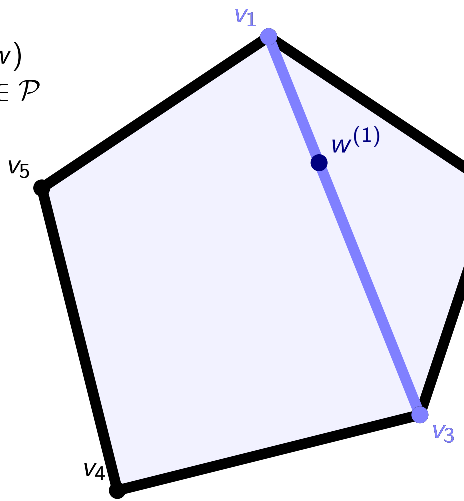
Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



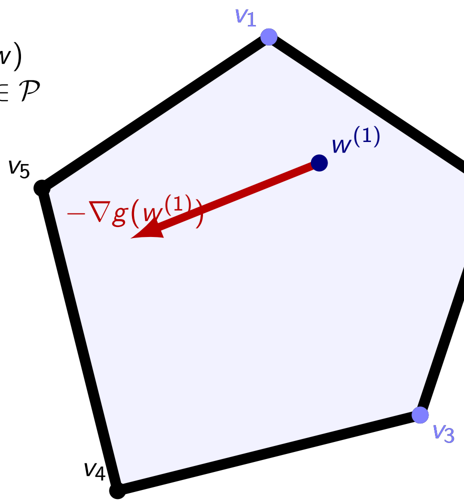
Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



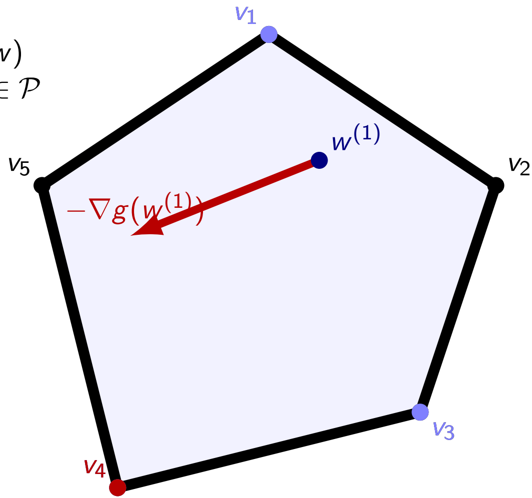
Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



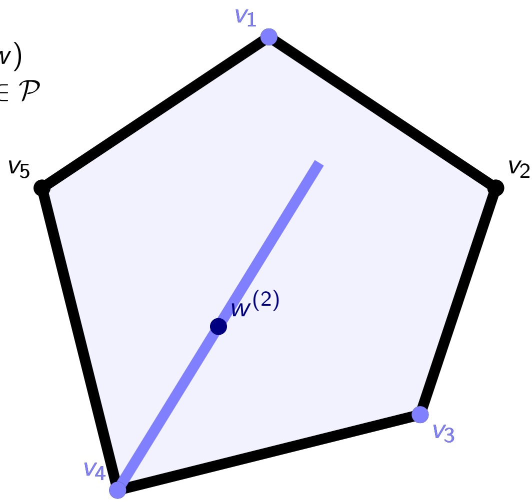
Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



Conditional gradient method (Frank-Wolfe)

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



What's wrong with CGM?

- slow
- complexity of iterate grows with number of iterations

Outline

Limited Memory Kelley's Method

Submodularity primer

- **ground set** $V = \{1, \ldots, n\}$
- identify subsets of V with Boolean vectors $\in \{0,1\}^n$
- $F : \{0,1\}^n \to \mathbb{R}$ is **submodular** if

$$F(A \cup v) - F(A) \geq F(B \cup v) - F(B), \quad \forall A \subseteq B,\ v \in V$$

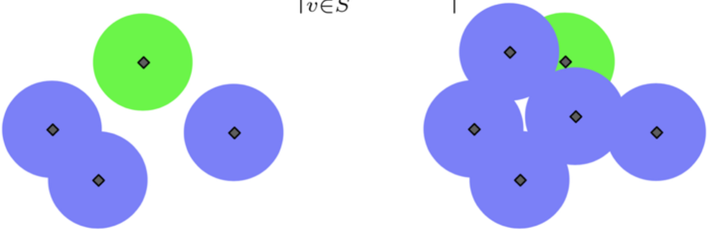
- linear functions are (sub)modular: for $w \in \mathbb{R}^n$, define

$$w(A) = \sum_{i \in A} w_i$$

Submodular function: example

Example: cover

$$F(S) = \left| \bigcup_{v \in S} \mathrm{area}(v) \right|$$



$$F(A \cup v) - F(A) \quad \geq \quad F(B \cup v) - F(B)$$

Submodular polyhedra

for submodular function F, define

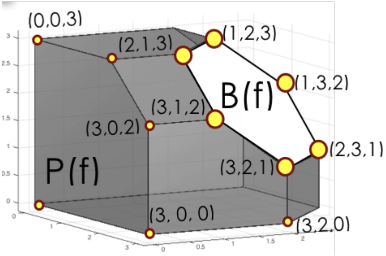
- **submodular polyhedron**

$$P(F) = \{w \in \mathbb{R}^n : w(A) \leq F(A), \quad \forall A \subseteq V\}$$

- **base polytope**

$$B(F) = \{w \in P(F) : w(V) = F(V)\}$$

both (generically) have exponentially many facets!



Lovász extension

define the (piecewise-linear) **Lovász extension** as

$$f(x) = \max_{w \in B(F)} w^\top x = \sup\{w(x) : w(A) \leq F(A) \forall A \subseteq V\}$$

the Lovász extension is the convex envelope of F

examples:

$F(A)$	$f(x)$	$f(\lvert x \rvert)$
$\lvert A \rvert$	$1^\top x$	$\lVert x \rVert_1$
$\min(\lvert A \rvert, 1)$	$\max(x)$	$\lVert x \rVert_\infty$
$\sum_{i=1}^{j} \min(\lvert A \cap S_j \rvert, 1)$	$\sum_{i=1}^{j} \max(x_{S_j})$	$\sum_{i=1}^{j} \lVert x_{S_j} \rVert_\infty$

Linear optimization on $B(F)$ is easy

- define the (piecewise-linear) **Lovász extension** as

$$f(x) = \max_{w \in B(F)} x^\top w$$

- f and $\mathbf{1}_{B(F)}$ are Fenchel duals:

$$f(x) = \mathbf{1}^*_{B(F)}(x)$$

- linear optimization over $B(F)$ is $O(n \log n)$ (Edmonds 1970)
  - define permutation π so $x_{\pi_1} \geq \cdots \geq x_{\pi_n}$. then

$$\max_{w \in B(F)} x^\top w = \sum_{k=1}^{n} x_{\pi_k} [F(\{\pi_1, \pi_2, \ldots, \pi_k\}) - F(\{\pi_1, \pi_2, \ldots, \pi_{k-1}\})]$$

- computing subgradients of f require $O(n \log n)$ too!

$$\partial f(x) = \operatorname*{argmax}_{w \in B(F)} x^\top w$$

Primal problem

$$\text{minimize} \quad g(x) + f(x) \qquad (\mathcal{P})$$

- $g : \mathbb{R}^n \to \mathbb{R}$ strongly convex
- $f : \mathbb{R}^n \to \mathbb{R}$ Lovász extension of submodular F
    - piecewise linear
    - homogeneous
    - (generically) exponentially many pieces
    - subgradients are easy $O(n \log n)$

Primal problem: example

Application to background subtraction

(Mairal, Jenatton, Obozinski, and Bach, 2010)

Input | ℓ_1-norm | Structured norm

(Source: `http://mistis.inrialpes.fr/learninria/slides/Bach.pdf`)

Original Simplicial Method (OSM) (Bach 2013)

Algorithm 1 OSM (to minimize $g(x) + f(x)$)

initialize $\mathcal{V} \leftarrow \emptyset$. repeat

1. define $\hat{f}(x) = \max_{w \in \mathcal{V}} w^\top x$
2. solve subproblem

$$x \leftarrow \operatorname{argmin} g(x) + \hat{f}(x)$$

3. compute $v \in \partial f(x) = \operatorname{argmax}_{w \in B(F)} x^\top w$
4. $\mathcal{V} \leftarrow \mathcal{V} \cup v$

problem:

- $\mathcal{V}$ keeps growing!
- No known rate of convergence (Bach 2013)

Limited Memory Kelley's Method ($\mathrm{LM\text{-}KM}$)

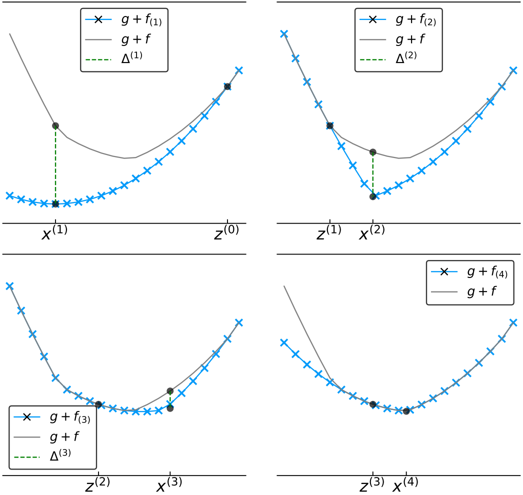
Algorithm 2 $\mathrm{LM\text{-}KM}$ (to minimize $g(x) + f(x)$)

initialize $\mathcal{V} \leftarrow \emptyset$. repeat

1. define $\hat{f}(x) = \max_{w \in \mathcal{V}} w^\top x$
2. solve subproblem
$$x \leftarrow \operatorname{argmin} g(x) + \hat{f}(x)$$
3. compute $v \in \partial f(x) = \operatorname{argmax}_{w \in B(F)} x^\top w$
4. $\mathcal{V} \leftarrow \{w \in \mathcal{V} : w^\top x = f(x)\} \cup v$

- does it converge or cycle?
- how large could $|\mathcal{V}|$ grow?

LM-KM: **intuition**



$\mathrm{L\text{-}KM}$ converges linearly with bounded memory

Theorem ((Zhou Gupta Udell 2018))

- $\mathrm{L\text{-}KM}$ *has bounded memory:* $|\mathcal{V}| \leq n + 1$
- $\mathrm{L\text{-}KM}$ *converges when g is strong convex*
- $\mathrm{L\text{-}KM}$ *converges linearly when g is smooth and strongly convex*

(Corollary: OSM converges linearly, too.)

Dual problem

$$\begin{array}{ll} \text{minimize} & -g^*(-w) \\ \text{subject to} & w \in B(F) \end{array} \qquad (\mathcal{D})$$

- $g^* : \mathbb{R}^n \to \mathbb{R}$ smooth (conjugate of strongly convex g)
- $B(F)$ base polytope of submodular F
  - (generically) exponentially many facets
  - linear optimization over $B(F)$ is easy $O(n \log n)$

Conditional gradient methods for the dual

- linear optimization over constraint is easy

so use a conditional gradient method!

- away-step FW, pairwise FW, fully corrective FW (FCFW) all converge linearly (Lacoste-Julien & Jaggi 2015)
- FCFW has limited memory
- (Garber & Hazan 2015) gives linear convergence with one gradient $+$ one linear optimization per iteration

Dual to primal

suppose g is α-strongly convex and β-smooth

- solve a dual subproblem inexactly to obtain $\hat{y} \in B(F)$ with

$$|g^*(-y^*) - g^*(-\hat{y})| \leq \epsilon$$

- g^* is $1/\beta$-strongly convex, so

$$\|\hat{y} - y^*\|^2 \leq 2\beta\epsilon$$

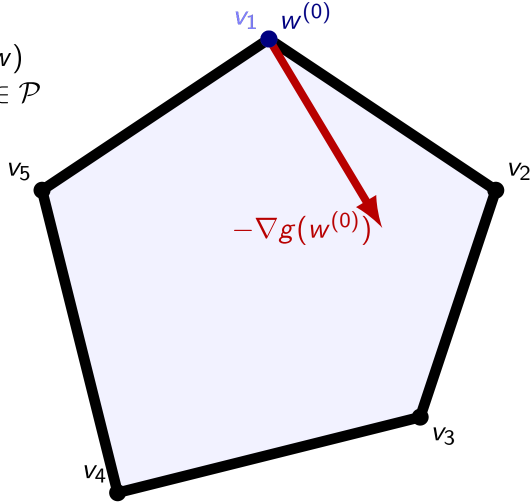
- define $\hat{x} = \nabla_y(-g^*(-\hat{y})) = \mathrm{argmin}_x\, g(x) + \hat{y}^\top x$
- since g^* is $1/\alpha$ smooth, we have

$$\|\hat{x} - x^\star\|^2 \leq 1/\alpha^2 \|\hat{y} - y^\star\|^2 \leq 2\beta\epsilon/\alpha^2$$

if the dual iterates converge linearly, so do the primal iterates

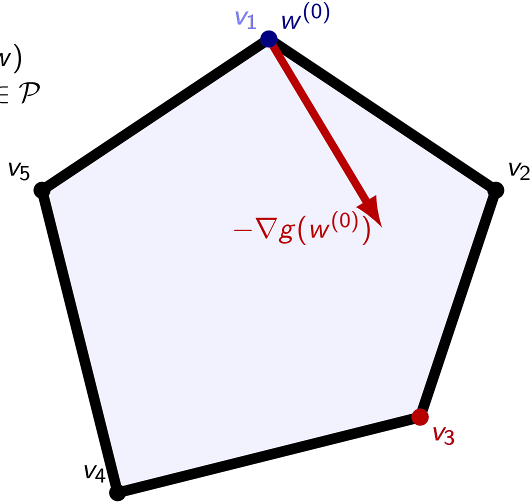
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



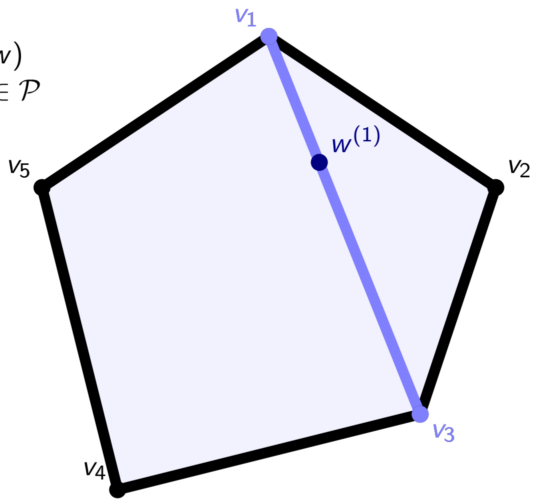
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



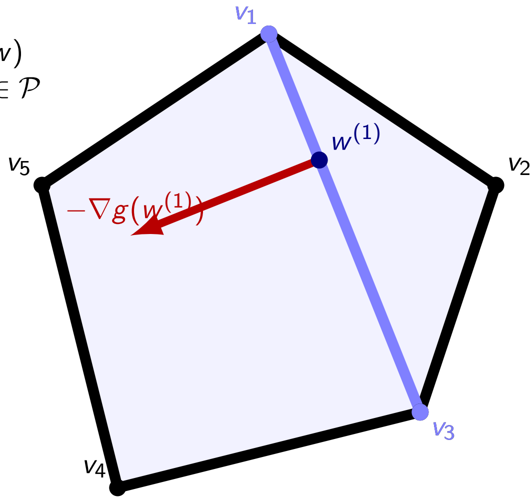
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



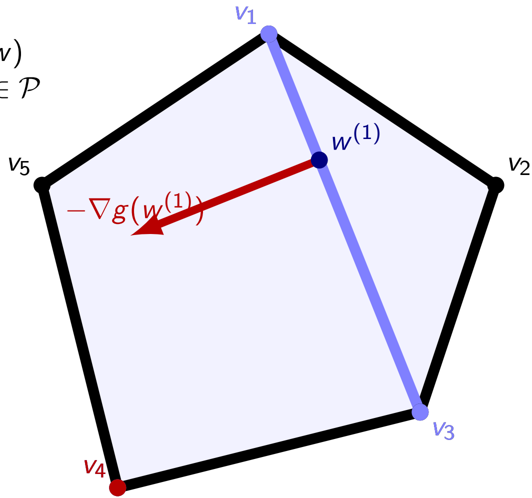
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



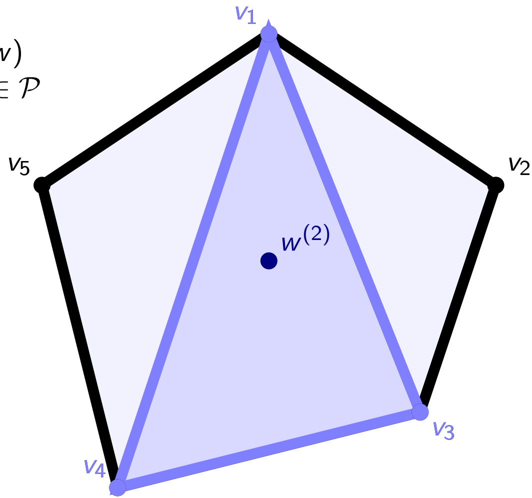
Fully corrective Frank-Wolfe

minimize $\quad g(w)$
subject to $\quad w \in \mathcal{P}$



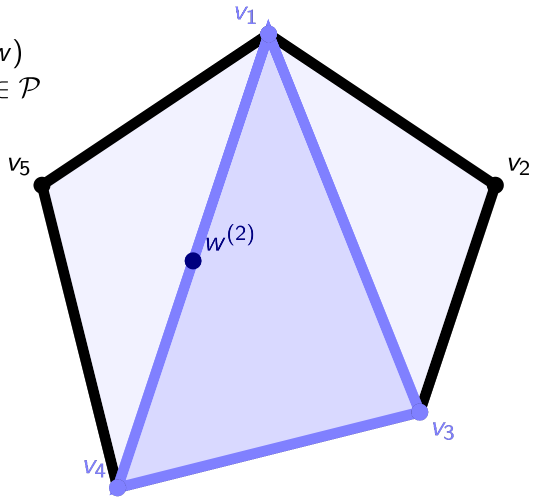
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



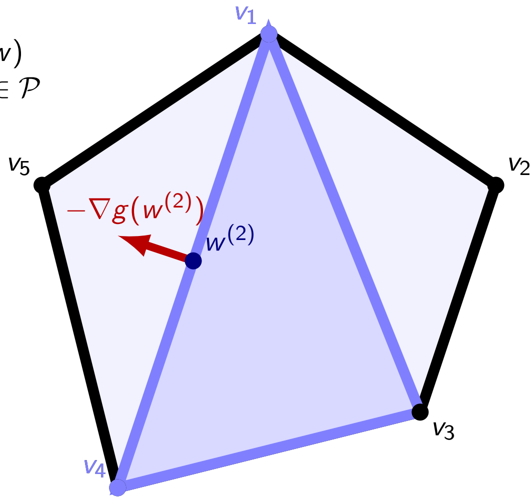
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



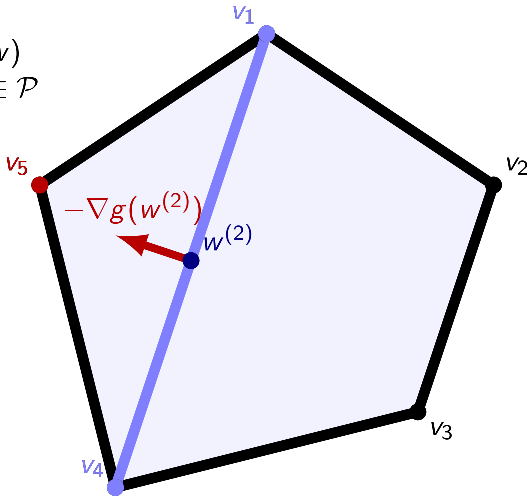
Fully corrective Frank-Wolfe

minimize $\quad g(w)$
subject to $\quad w \in \mathcal{P}$



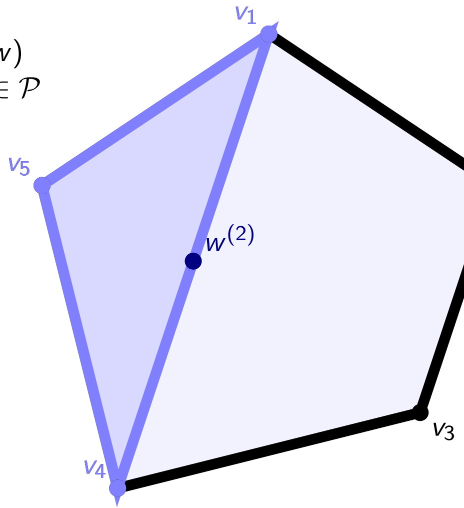
Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



Fully corrective Frank-Wolfe

$$\begin{array}{ll} \text{minimize} & g(w) \\ \text{subject to} & w \in \mathcal{P} \end{array}$$



Limited-memory Fully Corrective Frank Wolfe

L-FCFW

Algorithm 3 FCFW (to minimize $-g^*(-y)$ over $y \in B(F)$)

initialize $\mathcal{V} \leftarrow \emptyset$. repeat

1. solve subproblem

$$\begin{array}{ll} \text{minimize} & -g^*(-y) \\ \text{subject to} & y \in \textbf{Conv}(\mathcal{V}) \end{array}$$

   define solution $y = \sum_{w \in \mathcal{V}} \lambda_w w$
   with $\lambda_w > 0$ and $\sum_{w \in \mathcal{V}} \lambda_w = 1$
2. compute gradient $x = \nabla(-g^*(-y))$
3. solve linear optimization $v = \operatorname{argmax}_{w \in B(F)} x^\top w$
4. $\mathcal{V} \leftarrow \{w \in \mathcal{V} : \lambda_w > 0\} \cup v$

Fully corrective Frank Wolfe FCFW: properties

- **bounded memory:** Carathéodory $\implies$ can choose λ_w so

$$|\{w \in \mathcal{V} : \lambda_w > 0\}| \leq n + 1$$

- **finite convergence**
  - active set changes at each iteration
  - a vertex that exits the active set is never added again
- converges linearly for smooth strongly convex objectives
- useful if linear optimization over $B(F)$ is hard
  (so convex subproblem is comparatively cheap)
- ok to solve subproblem inexactly (Lacoste-Julien & Jaggi 2015)

compare to vanilla CGM:

- memory cost similar: $w \in \mathbb{R}^n$ vs n (very simple) vertices
- solving subproblems not much harder than evaluating g^*

Dual subproblems

- FCFW subproblem

$$\begin{array}{ll} \text{maximize} & -g^*(-y) \\ \text{subject to} & y = \sum_{w \in \mathcal{V}} \lambda_w w \\ & 1^T \lambda = 1, \ \ \lambda \geq 0 \end{array}$$

  has dual

$$\text{minimize} \quad g(x) + \max_{w \in \mathcal{V}} x^\top w$$

  which is our primal subproblem!

- first order optimality conditions show active sets match

$$\lambda_w > 0 \iff w^\top x = \max_{w \in \mathcal{V}} x^\top w$$

hence FCFW has a corresponding primal algorithm: $\mathrm{LM\text{-}KM}$!

LM-KM: numerical experiment

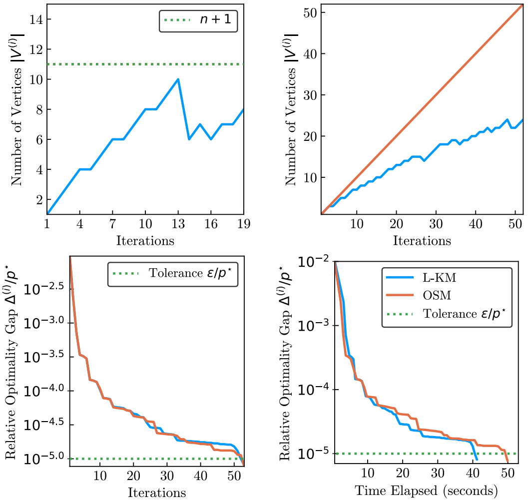
- $g(x) = x^\top Ax + b^\top x + n\|x\|^2$ for $x \in \mathbb{R}^n$
- f is the Lovász extension of

$$F(A) = \frac{|A|(2n - |A| + 1)}{2}$$

- entries of $A \in M_n$ sampled uniformly from $[-1, 1]$
- entries of $b \in \mathbb{R}^n$ sampled uniformly from $[0, n]$

LM-KM: **numerical experiment**

dimension $n = 10$ in upper left, $n = 100$ in others



Conclusion

$\mathrm{LM\text{-}KM}$ gives a new algorithm for composite convex and submodular optimization with **bounded ($\mathcal{O}(n)$) storage** with two old ideas:

- Duality
- Carathéodory

References

- S. Zhou, S. Gupta, and M. Udell. Limited Memory Kelley's Method Converges for Composite Convex and Submodular Objectives. NIPS 2018.