Generalised Self-concordant analysis of Frank-Wolfe algorithms

Pavel Dvurechensky[1] Shimrit Shtern[2] Mathias Staudigl[3]

[1]WIAS [2]The Technion [3]Maastricht University

Maastricht University

OWOS, April, 11, 2022

Ill-Conditioned minimisation in Machine Learning

- A common problem in machine learning is the minimisation of a convex function

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \ell_i(x) + \frac{\mu}{2}\|x\|_2^2.$$

- $\ell_i : \mathbb{R}^n \to \mathbb{R}$ is a statistical loss function (smooth)
- Typically n and m are *huge*.
- first-order (i.e. gradient based) methods are favorable optimization tools.
- Convergence rates depend on the *condition number* L_f/μ, where L_f is the Lipschitz modulus of ∇f.

A point for Ill-conditioned problems

Convex Lipschitz continuous losses lead to the general non-asymptotic bounds for the excess risk

$$f(x^*) - f(x^\circ) \approx \frac{L_f^2}{\mu m} + \mu \left\| x^\circ \right\|^2 = \text{Variance} + \text{Bias}.$$

Statistical optimal choice of the regularisation parameter $\mu = O(\frac{1}{\sqrt{m}})$.

If $m \gg 1$, then μ is very small. L_f is typically very large.

Optimization problems with large condition number are nearly ill-conditioned. A class of ill-conditioned problems which are tractable, are generalised self-concordant functions.

Problem Formulation

$\mathcal{X} \subset \mathsf{E}$ convex compact. Consider the optimisation problem

$$\min_{x \in \mathcal{X}} f(x). \tag{P}$$

Definition ([Sun and Tran-Dinh, 2018])

$f \in \mathbf{C}^3(\mathrm{dom}(f))$ with $\mathrm{dom}\, f$ open, is generalised self-concordant (GSC) if $\exists (M, \nu) \in \mathbb{R}_+ \times \mathbb{R}_+$ such that

$$|\varphi'''(t)| \leq M \varphi''(t)^{\nu/2}$$

for $\varphi(t) = f(x + td), x \in \mathrm{dom}\, f, d \in \mathsf{E}$ and $x + td \in \mathrm{dom}\, f$. Call $\mathcal{F}_{M_f,\nu}(\mathrm{dom}\, f)$ the set of GSC functions.

Cf. [Nesterov and Nemirovski, 1994, Bach, 2010, Tran-Dinh et al., 2019, Ostrovskii and Bach, 2021, Marteau-Ferey et al., 2019]

Generalised Self-concordant functions

- **Logistic Loss**

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \ln\left(1 + \exp(b_i \langle a_i, x\rangle)\right) + \frac{\mu}{2}\|x\|_2^2 .$$

where $b_i \in \{-1, 1\}, \mu > 0, a_i \in \mathbb{R}^n$.

- **Robust regression**

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \varphi(b_i - \langle a_i, x\rangle),\ \varphi(u) = \ln(e^u + e^{-u}).$$

- **Distance-Weighted Discrimination**

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} (a_i^\top w + \beta y_i + \xi_i)^{-q} + \langle c, \xi\rangle\ ,\ x = (w, \beta, \xi).$$

Self-concordant functions

- **Portfolio Optimisation**

$$f(x) = -\sum_{t=1}^{T} \ln(\langle r_t, x\rangle), x \in \mathcal{X} = \Delta_n$$

- **Covariance Estimation:**

$$f(x) = -\ln(\det(x)) + \mathrm{tr}(\Sigma x),$$
$$x \in \mathcal{X} = \{x \in \mathcal{S}_+^n : \|\mathrm{vec}(x)\|_1 \leq R\}.$$

- **Poisson Inverse Problem**

$$f(x) = \sum_{i=1}^{m} \langle w_i, x\rangle - \sum_{i=1}^{m} y_i \ln(\langle w_i, x\rangle),$$
$$x \in \mathcal{X} = \{x \in \mathbb{R}^n | \|x\|_1 \leq R\}.$$

Further applications

- **D-Optimal Design** Given m points $a_1, \dots, a_m \in \mathbb{R}^n$ whose affine hull is $\mathbb{R}^n$, find

$$\min f(x) = -\log\det\left(\sum_{i=1}^{m} x_i a_i a_i^\top\right) \text{ s.t.: } x \in \Delta_m.$$

- **Finding the analytic centre** Consider a domain $\{x \in \mathbb{R}^n | Ax \leq \mathbf{1}, x \in \{0,1\}^n\}$. Find an approximate feasible point by solving the analytic centre problem for the barrier

$$f(x) = -\log\left[L - \log\left(\sum_i \exp(L\langle a_i, x\rangle)\right)\right] - \left(\frac{2L}{3}\right)^2 \sum_{i=1}^{n} \log(x_i).$$

Standing Hypothesis

The following assumptions shall be in place:

(A.1) $f \in \mathcal{F}_{M_f,\nu}$ with $\nu \in [2,3]$.
(A.2) $\mathcal{X}^* = \operatorname{argmin}\{f(x) | x \in \mathcal{X}\} \neq \varnothing$:
(A.3) $\mathcal{X}$ is a convex compact subset in $\mathbb{R}^n$
(A.4) $\nabla^2 f$ is continuous and positive definite on $\operatorname{dom} f \cap \mathcal{X}$.

Conditional Gradient aka Frank-Wolfe

The analysis of FW involves

(a) a search direction

$$s(x) = \operatorname*{argmin}_{s \in \mathcal{X}} \langle \nabla f(x), s \rangle .$$

(b) as merit function

$$\mathrm{Gap}(x) = \langle \nabla f(x), x - s(x) \rangle$$

Standard Frank-Wolfe method:

If $\mathrm{Gap}(x^k) > \varepsilon$ then

1. Obtain $s^k = s(x^k)$;
2. Update $x^{k+1} = x^k + \alpha_k (s^k - x^k)$ for some $\alpha_k \in [0, 1)$.

Why projection-free optimization?

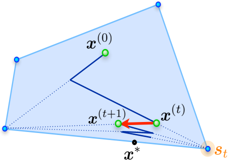
- First-order methods in covex optimization gained significance in connection with large-scale optimization problems.
- Optimization models are dependent on data that can be noisy, so no need for high-accuracy solutions.
- First-order methods are appealing in practice because of their lower computational burden per iteration.
- First-order methods are able to preserve problem structure (e.g. sparsity), and can be extended to non-smooth problems.

See [Dvurechensky et al., 2021] for a recent survey.

Convergence Analysis

Because of great scalability and sparsity properties, *Frank-Wolfe* (FW) methods (Frank & Wolfe, 1956) received lot of attention in ML.

1. Convergence guarantees require Lipschitz continuous gradients, or finite curvature constants on f (Jaggi, 2013)
2. Even for well-conditioned problems only sublinear convergence rates guaranteed in general.



Why do standard Methods fail in ill-conditioned problems?

Consider $f(x_1, x_2) = -\ln(x_1) - \ln(x_2)$ over $x_1, x_2 \in [0, 1], x_1 + x_2 = 1$.

- Start from $x^0 = (1/4, 3/4)$
- Apply the standard $2/(k+2)$-step size policy, then $\alpha_0 = 1$.
- $x^1 = (1, 0) \notin \operatorname{dom} f$.

The Dikin Ellipsoid

- The analysis of GSC minimisation algorithms makes use of the local norm:

$$\|a\|_x \triangleq \sqrt{\langle \nabla^2 f(x) a, a \rangle}, \|a\|_x^* \triangleq \sqrt{\langle a, [\nabla^2 f(x)]^{-1} a \rangle}$$

for $x \in \operatorname{dom} f$.

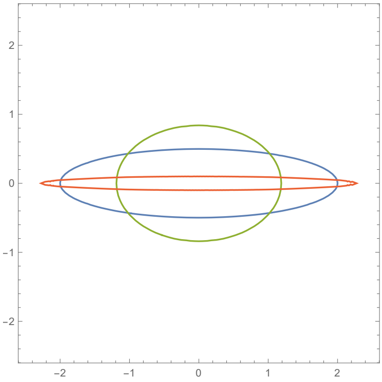
- Define the metric

$$\mathrm{d}_\nu(x, y) \triangleq \begin{cases} M_f \|y - x\|_2 & \text{if } \nu = 2, \\ \frac{\nu-2}{2} M_f \|y - x\|_2^{3-\nu} \|y - x\|_x^{\nu-2} & \text{if } \nu > 2. \end{cases}$$

Dikin Ellipsoid

The Dikin Ellipsoid is defined as

$$\mathcal{W}(x, r) \triangleq \{y \in \mathsf{E} | \mathrm{d}_\nu(x, y) < r\} \subset \mathrm{dom}\, f \quad \forall r \in (0, 1).$$



Algorithm 1: Analytic step size method

Algorithm `FW-GSC`

Input: $x^0 \in \text{dom } f \cap \mathcal{X}$ initial state, $\varepsilon > 0$ error tolerance, and $f \in \mathcal{F}_{M,\nu}(\text{dom } f)$.

for $k = 0, \ldots$ **do**
  if $\text{Gap}(x^k) > \varepsilon$ **then**
    Obtain $s^k = s(x^k)$
    Obtain $\alpha_k = \alpha_{M_f,\nu}(x^k)$
    Set $x^{k+1} = x^k + \alpha_k(s^k - x^k)$
  end if
end for

Derivations

Let $x_t^+ = x + t(s(x) - x), t > 0$
For $t > 0$ such that $\mathrm{d}_\nu(x, x_t^+) < 1$, obtain the GSC descent inequality:

$$f(x_t^+) \leq f(x) + \langle \nabla f(x), x_t^+ - x \rangle + \omega_\nu(\mathrm{d}_\nu(x, x_t^+)) \left\| x_t^+ - x \right\|_x^2$$

Optimising the per-iteration decrease w.r.t t leads to an analytic step-size criterion

$$\alpha_{M_f,\nu}(x) = \min\{1, \mathtt{t}_{M_f,\nu}(x)\}.$$

Algorithm 1

Determining the step size

The GSC descent lemma can be written as

$$f(x_t^+) \leq f(x) - t\mathrm{Gap}(x) + \omega_\nu(tM_f\delta_\nu(x))t^2 \mathrm{e}(x)^2, \\ = f(x) - \eta_{x,M_f,\nu}(t) \qquad t \in (0, 1/\delta_\nu(x)),$$

where

$$\omega_\nu(t) = \begin{cases} \frac{1}{t^2}(e^t - t - 1) & \text{if } \nu = 2, \\ \frac{-t-\ln(1-t)}{t^2} & \text{if } \nu = 3, \\ \left(\frac{\nu-2}{4-\nu}\right)\frac{1}{t}\left[\frac{\nu-2}{2(3-\nu)t}((1-t)^{\frac{2(3-\nu)}{2-\nu}} - 1) - 1\right] & \text{if } \nu \in (2,3). \end{cases}$$

$$\delta_\nu(x) = \begin{cases} \beta(x) & \text{if } \nu = 2, \\ \frac{\nu-2}{2}\beta(x)^{3-\nu}\mathrm{e}(x)^{\nu-2} & \text{if } \nu > 2, \end{cases} \quad \text{and}$$

$$\beta(x) = \|s(x) - x\|_2, \quad \mathrm{e}(x) = \|s(x) - x\|_x,$$

$$\eta_{x,M,\nu}(t) = \mathrm{Gap}(x)\left[t - \omega_\nu(tM\delta_\nu(x))t^2\frac{\mathrm{e}(x)^2}{\mathrm{Gap}(x)}\right].$$

Solving $\max_t \eta_{x,\nu}(t)$ yields

$$\mathtt{t}_{M,\nu}(x) = \begin{cases} \frac{1}{M\delta_2(x)} \ln\left(1 + \frac{\mathrm{Gap}(x) M\delta_2(x)}{\mathrm{e}(x)^2}\right) & \text{if } \nu = 2, \\ \frac{1}{M\delta_\nu(x)}\left[1 - \left(1 + \frac{M\delta_\nu(x)\mathrm{Gap}(x)}{\mathrm{e}(x)^2}\frac{4-\nu}{\nu-2}\right)^{-\frac{\nu-2}{4-\nu}}\right] & \text{if } \nu \in (2,3) \\ \frac{\mathrm{Gap}(x)}{M\delta_3(x)\mathrm{Gap}(x)+\mathrm{e}(x)^2} & \text{if } \nu = 3. \end{cases}$$

Calling $\Delta_k = \eta_{x^k,\nu}(\alpha_\nu(x^k))$, to get

$$f(x^{k+1}) \leq f(x^k) - \Delta_k < f(x^k)$$

Asymptotic Convergence

Proposition

Let $(x^k)_{k \in \mathbb{N}_0}$ be generated by algorithm `FW-GSC`*. Then the following assertions hold:*

1. $\left(f(x^k)\right)_k$ *is non-increasing;*
2. $\sum_k \Delta_k < \infty$ *and hence* $\lim_{k \to \infty} \Delta_k = 0$;
3. *For all* $K \geq 1$ *we have* $\min_{k \leq K} \Delta_k \leq \frac{1}{K}(f(x^0) - f(x^*))$.

Iteration Complexity

Define the approximation error : $h_k = f(x^k) - f^*$. Let

$$S(x^0) = \{x \in \mathcal{X} | f(x) \leq f(x^0)\}, \text{ and}$$
$$L_{\nabla f} = \max_{x \in S(x^0)} \lambda_{\max}(\nabla^2 f(x)).$$

Theorem

For given $\varepsilon > 0$, define $N_\varepsilon(x^0) = \min\{k \geq 0 | h_k \leq \varepsilon\}$. Then,

$$N_\varepsilon(x^0) \leq \frac{\ln\left(\frac{c_1(M_f,\nu)}{h_0 c_2(M_f \nu)}\right)}{\ln(1 - c_1(M_f, \nu))} + \frac{1}{c_2(M_f, \nu)\varepsilon},$$

where $c_1(M_f, \nu)$, $c_2(M_f, \nu)$ are explicit constants.

An adaptive quadratic model-based algorithm

Consider the quadratic model

$$Q(x, t, \mathcal{L}) = f(x) - t\mathrm{Gap}(x) + \frac{t^2\mathcal{L}}{2}\|s(x) - x\|_2^2 .$$

On the level set $S(x^k)$, we get the descent lemma

$$f(x^k + t(s(x^k) - x^k)) \leq Q(x^k, t, L_k)$$

for L_k a local estimate of the Lipschitz constant on the level set.
A backtracking strategy on L_k yields a new algorithm.

Algorithm 2

Algorithm LBTFWGSC

Input: $x^0 \in \operatorname{dom} f \cap \mathcal{X}$ initial state; $\mathcal{L}_{-1}$ initial Lipschitz estimate, $\gamma_u > 1 > \gamma_d$.
for $k = 1, \ldots$ **do**
  if $\operatorname{Gap}(x^k) > \varepsilon$ **then**
    Obtain $s^k = s(x^k)$ and set $v^k = s^k - x^k$
    Set $(\alpha_k, \mathcal{L}_k) = \mathtt{step}_L(f, v^k, x^k, \mathcal{L}_{k-1})$
    Set $x^{k+1} = x^k + \alpha_k(s^k - x^k)$
  end if
end for

Algorithm Function $\mathtt{step}_L(f, v, x, \mathcal{L})$

Choose $\bar{L} \in [\gamma_d \mathcal{L}, \mathcal{L}]$
$\alpha = \min\{\frac{\operatorname{Gap}(x)}{\bar{L}\|v\|_2^2}, 1\}$
if $x + \alpha v \notin \operatorname{dom} f$ or $f(x + \alpha v) > Q_L(x, \alpha, \bar{L})$ **then**
  $\bar{L} \leftarrow \gamma_u \bar{L}$
  $\alpha \leftarrow \min\{\frac{\operatorname{Gap}(x)}{\bar{L}\|v\|_2^2}, 1\}$
end if
Return $\alpha, \bar{L}$

Complexity Estimate

Theorem

Let $(x^k)_k$ be generated by `LBTFWGSC`. *Then*

$$N_\varepsilon(x^0) \leq \frac{2\bar{L}\,\mathrm{diam}(\mathcal{X})^2}{\varepsilon} + \frac{\ln(\bar{L}\,\mathrm{diam}(\mathcal{X})^2/h_0)}{\ln(1/2)}$$

where $\bar{L} = \max\{\gamma_u L_{\nabla f}, \mathcal{L}_{-1}\}$.

Searching for the scale parameter

Let $v_{FW}(x) = s(x) - x$ the FW-search direction.

Suppose $\mu > 0$ is a local guess of the GSC parameter M_f. For the search point $x_t^+ = x + tv_{FW}(x^k)$ we have

$$f(x_t^+) \leq f(x) - t\mathrm{Gap}(x) + t^2 \mathrm{e}(x)^2 \omega_\nu(t\mu\delta_\nu(x)) \equiv Q_M(x, t, \mu).$$

Optimize the new model with respect to t gives a step size policy $\alpha_\nu(x, \mu)$.

Search for the best μ to obtain a close fit between the upper model and the actual function values.

Algorithm 3: MBTFWGSC

Algorithm MBTFWGSC

Input: $x^0 \in \mathrm{dom}\, f \cap \mathcal{X}$ initial state; μ_{-1} initial Lipschitz estimate, $\gamma_u > 1 > \gamma_d$.
for $k = 1, \ldots$ **do**
  if $\mathrm{Gap}(x^k) > \varepsilon$ **then**
    Obtain $s^k = s(x^k)$ and set $v^k = s^k - x^k$
    Set $(\alpha_k, \mu_k) = \mathtt{step}_M(f, v^k, x^k, \mu_{k-1})$
    Set $x^{k+1} = x^k + \alpha_k v^k$
  end if
end for

Algorithm Function $\mathtt{step}_M(f, v, x, \mu)$

Choose $\bar{M} \in [\gamma_d \mu, \mu]$
$\alpha = \alpha_{\bar{M},\nu}(x)$
if $x + \alpha v \notin \mathrm{dom}\, f$ or $f(x + \alpha v) > Q_M(x, \alpha, \bar{M})$ **then**
  $\bar{M} \leftarrow \gamma_u \bar{M}$
  $\alpha \leftarrow \alpha_{\bar{M},\nu}(x)$
end if
Return $\alpha, \bar{M}$

Complexity Analysis

Theorem

Let $(x^k)_k$ be generated by MBTFWGSC. *Then*

$$N_\varepsilon(x^0) \leq \frac{\ln\left(\frac{c_1(\tilde{M},\nu)}{h_0 c_2(\tilde{M},\nu)}\right)}{\ln(1 - c_1(\tilde{M},\nu))} + \frac{1}{c_2(\tilde{M},\nu)\varepsilon},$$

where $\tilde{M} = \max\{\gamma_u M_f, \mu_{-1}\}$.

Preparations

- All methods so far displayed a complexity of $O(1/\varepsilon)$;
- It is known that FW can be accelerated under various hypothesis:
  - Strong convexity coupled with interior solutions [GuéLat and Marcotte, 1986, Lacoste-Julien and Jaggi, 2015];
  - Composition of strongly convex with affine transformation [Beck and Shtern, 2017];
  - $\mathcal{X}$ strongly convex [Garber and Hazan, 2015, Kerdreux and d'Aspremont, 2020];

  see the recent survey [Bomze et al., 2021].

Assumption

$$\mathcal{X} = \{x \in \mathbb{R}^n | Bx \leq b\}$$

Local Linear minimization oracle

Definition ([Garber and Hazan, 2016])

A procedure $\mathcal{A}(x, r, c)$, where $x \in \mathcal{X}, r > 0, c \in \mathbb{R}^n$, is a LLOO with parameter $\rho \geq 1$ for the polytope $\mathcal{X}$ if $\mathcal{A}(x, r, c)$ returns a point $u = u(x, r, c) \in \mathcal{X}$ such that for all $x \in \mathbb{B}_r(x) \cap \mathcal{X}$

$$\langle c, x \rangle \geq \langle c, s \rangle \ \text{ and } \ \|x - s\|_2 \leq \rho r.$$

- Such oracles exist for any compact polyhedral domain.
- Particular simple implementation for Simplex-like domains.

Modifications

Define the modified merit function

$$\begin{aligned}\Gamma(x, r) &= \langle \nabla f(x), x - u(x, r, \nabla f(x)) \rangle \\ &= \max_{s \in \mathbb{B}_r(x) \cap \mathcal{X}} \langle \nabla f(x), x - s \rangle\end{aligned}$$

and

$$u(x, r, c) = \min_{s \in \mathbb{B}_r(x) \cap \mathcal{X}} \langle c, s \rangle .$$

Algorithm 4: FWLLOO

Algorithm FWLLOO

Input: $\mathcal{A}(x, r, c)$-LLOO with parameter $\rho \geq 1$ for polytope $\mathcal{X}$, $f \in \mathcal{F}_{M_f,\nu}(\text{dom } f)$. $\sigma_f > 0$ convexity parameter.
$x^0 \in \text{dom } f \cap \mathcal{X}$, and let $h_0 = f(x^0) - f^*$, and $c_0 = 1$.
for $k = 0, 1, \ldots,$ **do**
  Set $r_k = r_0^2 c_k$
  Obtain $u^k = u(x^k, r_k, \nabla f(x^k))$
  Set $\alpha_k = \alpha_\nu(x^k)$
  Update $x^{k+1} = x^0 + \alpha_k(u^k - x^k)$
end for

Iteration Complexity

Theorem

Let $(x^k)_{k\geq 0}$ be generated by `FWLLOO`. *Then, for all $k \geq 0$, we have $x^* \in \mathbb{B}_{r_k}(x^k)$ and*

$$h_k \leq \text{Gap}(x^0) \exp\left(-\frac{1}{2}\sum_{i=0}^{k-1} \alpha_i\right).$$

Preparations

- FWLLOO needs σ_f or $L_{\nabla f}$ as input.
- Both are hard to estimate in practice.
- Away step method exploits the geometry of $\mathcal{X}$, and a Hoffman bound to compensate for these input parameters.

Definition

Let $\mathcal{U} = \mathtt{Ext}(\mathcal{X})$, so that $\mathcal{X} = \mathrm{conv}(\mathcal{U})$. $\mu : \mathcal{U} \to [0, 1]$ is a vertex representration of x, if $x = \sum_{u \in \mathcal{U}} \mu_u u$. Let $\mathcal{U}(x)$ be the set of active vertices at x

Away Steps

Assumption ([Beck and Shtern, 2017])

The LMO is a vertex linear oracle:

$$s(x) \in \operatorname*{argmin}_{d \in \mathcal{X}} \langle \nabla f(x), d \rangle$$

returns a point in $\mathcal{U}$.

Definition

Given $x \in \mathcal{X}$, we call

- $v_{FW}(x) = s(x) - x$ a forward step
- $v_A(x) = x - u(x)$, where $u(x) \in \operatorname{argmax}_{d \in \mathcal{U}(x)} \langle \nabla f(x), d \rangle$, an away step.

Algorithm 5: ASFWGSC

Algorithm ASFWGSC

Input: $x^0 \in \operatorname{dom} f \cap \mathcal{U}$ where $\mu_u^1 = 0$ for all $u \in \mathcal{U} \setminus \{x^1\}$ and $U^1 = \{x^1\}$.
for $k = 0, 1, \ldots$ **do**
  Set $s^k = s(x^k), u^k = u(x^k)$, and $v_A(x^k) = x^k - u^k$, $v_{FW}(x^k) = s^k - x^k$
  if $\left\langle \nabla f(x^k), s^k - x^k \right\rangle \leq \left\langle \nabla f(x^k), x^k - u^k \right\rangle$ **then**
    Set $v^k = v_{FW}(x^k)$
  else
    Set $v^k = v_A(x^k)$
  end if
  Set $\beta_k = \left\| v^k \right\|_2, e_k = \left\| v^k \right\|_{x^k}, \bar{t}_k \equiv \bar{t}(x^k)$
  Find $\alpha_k = \operatorname{argmin}_{t \in [0, \bar{t}_k]} t \left\langle \nabla f(x^k), v^k \right\rangle + t^2 e_k^2 \omega_\nu(t M_f \delta_\nu(x^k))$
  Update $x^{k+1} = x^k + \alpha_k v^k$
  if $v^k = v_{FW}(x^k)$ **then**
    Update $U^{k+1} = U^k \cup \{s^k\}$
  else
    if $v^k = v_A(x^k)$ and $\alpha_k = \bar{t}_k$ **then**
      Update $U^{k+1} = U^k \setminus \{u^k\}$ and μ^{k+1}
    else
      Update $U^{k+1} = U^k$
    end if
  end if
end for

Iteration Complexity

Theorem

Let $\{x^k\}_{k\in\mathbb{N}}$ be the trajectory generated by ASFWGSC*. Then, for all $k \geq 0$, we have*

$$h_k \leq h_0 \exp(-\theta k/2).$$

where $\theta = \min\left\{0.5, \frac{c_1(M_f,\nu)\Omega}{2\,\mathrm{diam}(\mathcal{X})}, \frac{c_2(M_f,\nu)\Omega^2\sigma_f}{8}\right\}$.

The Elastic Net

Consider

$$f(x) = \frac{1}{p} \sum_{i=1}^{p} \log\left(1 + \exp(-y_i \langle a_i, x \rangle + \mu)\right) + \frac{\gamma}{2} \|x\|_2^2$$

Since [Bach, 2010], we know that this can be seen as a GSC minimization problem with $\nu = 2$ or $\nu = 3$.
Consider the elastic net formulation

$$\min_{x : \|x\|_1 \leq R} f(x).$$

Dependence on the GSC model

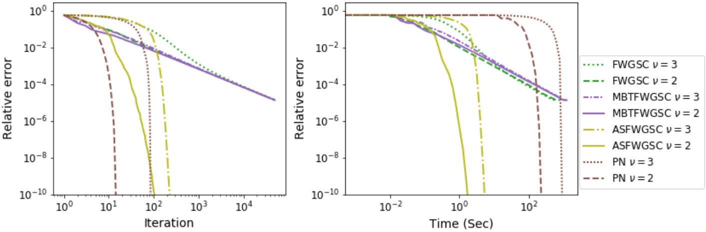
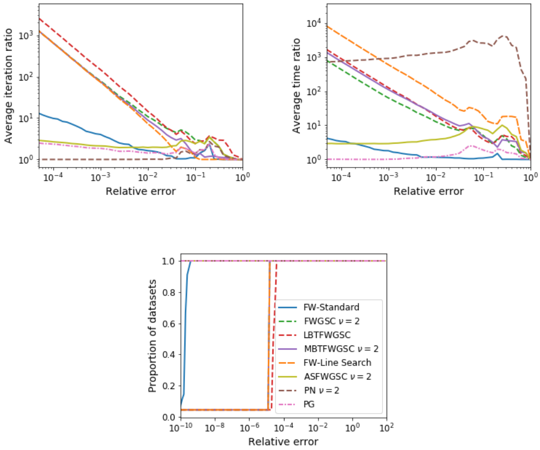


Figure: Comparison between $\nu = 3$ and $\nu = 2$ for data set a9a.

Numerical Results - Performance Profiles



Experimental Setup

Consider the distance weighted discrimination (DWD) problem, introduced in [Marron et al., 2007].

The classification loss attains the form

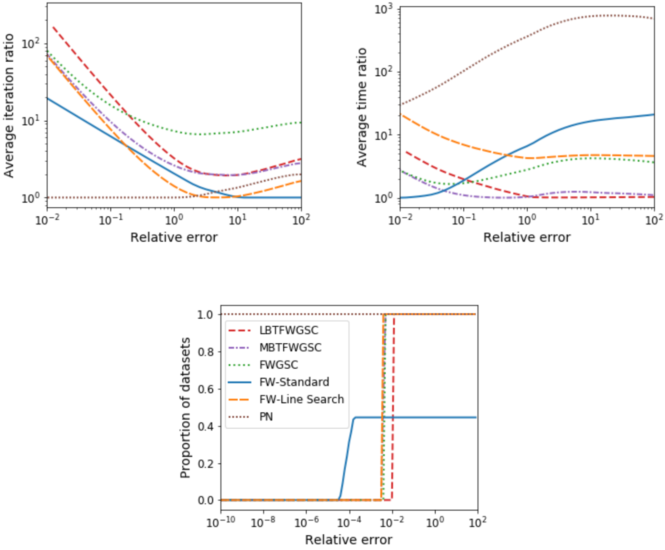
$$f(x) = \frac{1}{p}\sum_{i=1}^{p}(a_i^\top w + \mu y_i + \xi_i)^{-q} + c^\top \xi,$$

over the convex compact set

$$\mathcal{X} = \{x = (w, \mu, \xi) |\ \|w\|^2 \leq 1, \mu \in [-u, u], \|\xi\|^2 \leq R, \xi \in \mathbb{R}^p_+\},$$

where $R > 0$ is a hyperparameter that has to be learned via cross-validation.

Results on DWD



Experimental Setup

Consider learning a Gaussian graphical random field of p nodes/variables.

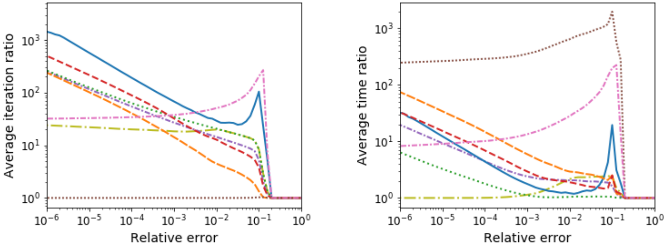
To learn the graphical model via an ℓ_1-regularization framework in its constrained formulation, we minimize the loss function

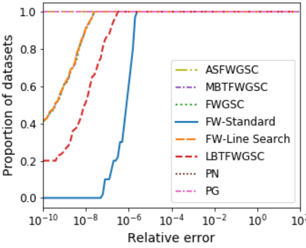
$$f(x) = -\log\det(\mathtt{mat}(x)) + \mathrm{tr}(\hat{\Sigma}\mathtt{mat}(x))$$

over

$$\mathcal{X} = \{x \in \mathbb{R}^n | \ \|x\|_1 \leq R, \mathtt{mat}(x) \in \mathcal{S}^n_+\}$$

Covariance Estimation





Thank you!

For details see:
[Dvurechensky et al., 2020, Dvurechensky et al., 2022]

References I

- Bach, F. (2010).
  Self-concordant analysis for logistic regression.
  Electron. J. Statist., 4:384–414.

- Beck, A. and Shtern, S. (2017).
  Linearly convergent away-step conditional gradient for non-strongly convex functions.
  Mathematical Programming, 164(1):1–27.

- Bomze, I. M., Rinaldi, F., and Zeffiro, D. (2021).
  Frank-wolfe and friends: a journey into projection-free first-order optimization methods.
  4OR, 19(3):313–345.

References II

Dvurechensky, P., Ostroukhov, P., Safin, K., Shtern, S., and Staudigl, M. (2020).
Self-concordant analysis of Frank-Wolfe algorithms.
In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2814–2824, Virtual. PMLR. arXiv:2002.04320.

Dvurechensky, P., Safin, K., Shtern, S., and Staudigl, M. (2022).
Generalized self-concordant analysis of frank-wolfe algorithms.
Mathematical Programming.

References III

Dvurechensky, P., Shtern, S., and Staudigl, M. (2021).
First-order methods for convex optimization.
EURO Journal on Computational Optimization, 9:100015.
arXiv:2101.00935.

Garber, D. and Hazan, E. (2015).
Faster rates for the frank-wolfe method over strongly-convex sets.
In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 541–549, Lille, France. PMLR.

References IV

Garber, D. and Hazan, E. (2016).
A linearly convergent variant of the Conditional Gradient algorithm under strong convexity, with applications to online and stochastic optimization.
SIAM Journal on Optimization, 26(3):1493–1528.

GuéLat, J. and Marcotte, P. (1986).
Some comments on Wolfe's 'away step'.
Mathematical Programming, 35(1):110–119.

Kerdreux, T. and d'Aspremont, A. (2020).
Frank-wolfe on uniformly convex sets.
arXiv preprint arXiv:2004.11053.

References V

Lacoste-Julien, S. and Jaggi, M. (2015).
On the global linear convergence of Frank-Wolfe optimization variants.
In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28, pages 496–504. Curran Associates, Inc.

Marron, J. S., Todd, M. J., and Ahn, J. (2007).
Distance-weighted discrimination.
Journal of the American Statistical Association, 102(480):1267–1271.

References VI

Marteau-Ferey, U., Ostrovskii, D., Bach, F., and Rudi, A. (2019).
Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance.
In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2294–2340, Phoenix, USA. PMLR.

Nesterov, Y. and Nemirovski, A. (1994).
Interior Point Polynomial methods in Convex programming.
SIAM Publications.

References VII

Ostrovskii, D. M. and Bach, F. (2021).
Finite-sample analysis of *m*-estimators using self-concordance.
Electronic Journal of Statistics, 15(1):326–391.

Sun, T. and Tran-Dinh, Q. (2018).
Generalized self-concordant functions: a recipe for Newton-type methods.
Mathematical Programming.

Tran-Dinh, Q., Sun, T., and Lu, S. (2019).
Self-concordant inclusions: a unified framework for path-following generalized Newton-type algorithms.
Mathematical Programming, 177(1):173–223.