HSE University 23 December

# Step-size strategies in the conditional gradient algorithm

Ilya Levin Pavel Zakharov Valerija Shcherbakova

#### Table of content

- Frank-Wolfe algorithm
- Predefined decreasing sequence
- Demyanov-Rubinov step-size
- Exact line-search
- Backtracking line-search
- Results

### Frank-Wolfe algorithm

is a method for constrained optimization that solves problems of the form  $\min_{m{x}\in\mathcal{D}} f(m{x})$ 

where f is a smooth function for which we have access to its gradient and D is a compact set. We also assume to have access to a *linear minimization oracle* over D, that is, a routine that solves problems of the form

$$oldsymbol{s}_t \in rg \max_{oldsymbol{s} \in \mathcal{D}} \langle -
abla f(oldsymbol{x}_t), oldsymbol{s} 
angle$$

## Frank-Wolfe algorithm pseudo code

#### Frank-Wolfe algorithm

Input: initial guess  $x_0$ , tolerance  $\delta > 0$ 

$$\textbf{For } t = 0, 1, \dots \textbf{ do } \qquad \boldsymbol{s}_t \in \argmax_{\boldsymbol{s} \in \mathcal{D}} \langle -\nabla f(\boldsymbol{x}_t), \boldsymbol{s} \rangle$$

Set 
$$oldsymbol{d}_t = oldsymbol{s}_t - oldsymbol{x}_t$$
 and  $oldsymbol{g}_t = \langle - 
abla f(oldsymbol{x}_t), oldsymbol{d}_t 
angle$ 

Choose step-size  $\gamma_t$  (to be discussed later)

$$oldsymbol{x}_{t+1} = oldsymbol{x}_t + oldsymbol{\gamma}_t oldsymbol{d}_t$$
 .

end For loop

return  $\boldsymbol{x}_t$ 

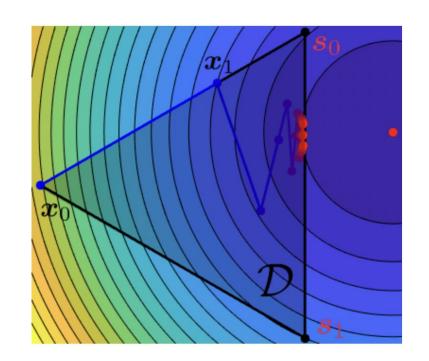
# Predefined decreasing sequence

$$oldsymbol{\gamma}_t = rac{2}{t+2}$$

is to choose the step-size according to the pre-defined decreasing sequence

- 1) is straightforward;
  - 2) cheap to compute.

- in practice it performs worse than the alternatives;
  - 2) Oscillates a lot.



# **Demyanov-Rubinov**

$$oldsymbol{\gamma} = \min\left\{rac{g_t}{oldsymbol{L}\|oldsymbol{d}_t\|^2}, 1
ight\}$$

is to choose the step-size according to the pre-defined decreasing sequence



- Not that expensive to compute;
- Good convergence in practice.

- 1) Can be problematic to estimate L;
  - 2) Convergence in the neighbourhood of solution is not optimal.

#### **Exact line-search**

$$oldsymbol{\gamma}_{\star} \in rg \min_{oldsymbol{\gamma} \in [0,1]} f(oldsymbol{x}_t + oldsymbol{\gamma} oldsymbol{d}_t)$$

takes the step-size that maximizes the decrease in objective along the update direction



- Great convergence in practice;
- Gives the highest decrease per iteration.

1) Pretty heavy computation if we don't have an access to the minimizer of f.

# **Backtracking line-search**

$$oldsymbol{\gamma}_t = \min\left\{rac{g_t}{oldsymbol{M}_t\|oldsymbol{d}_t\|^2}, 1
ight\}$$

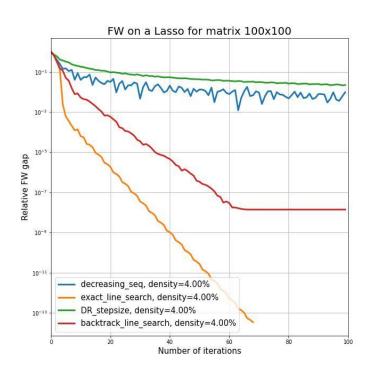
is to choose the step-size according to the pre-defined decreasing sequence

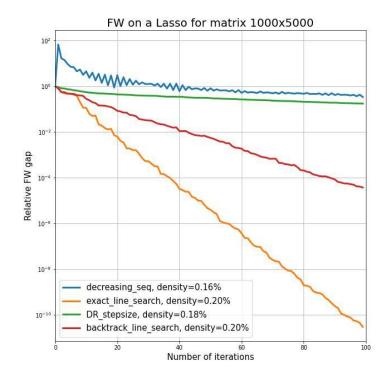
$$Q_t(oldsymbol{\gamma}, oldsymbol{M}_t) \stackrel{ ext{def}}{=} f(oldsymbol{x}_t) - oldsymbol{\gamma} g_t + rac{oldsymbol{\gamma}^2 oldsymbol{M}_t}{2} \|oldsymbol{d}_t\|^2$$

- wildly successful and are a core part of any state-of-the-art implementation of (proximal) gradient descent and Quasi-Newton methods.
- Highly depends on the input parameters

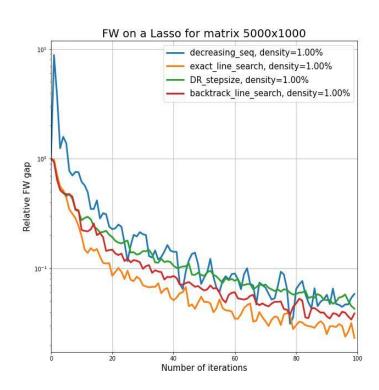
$$egin{aligned} m{M}_t &= m{\eta} m{M}_{t-1} \ m{\gamma}_t &= \min \left\{ g_t / (m{M}_t \| m{d}_t \|^2), 1 
ight\} \ ext{While } f(m{x}_t + m{\gamma}_t m{d}_t) > Q_t(m{\gamma}_t, m{M}_t) ext{ do } \ m{M}_t &= m{ au} m{M}_t \end{aligned}$$

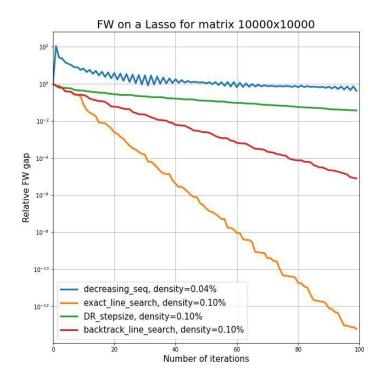
## Our results: linear regression



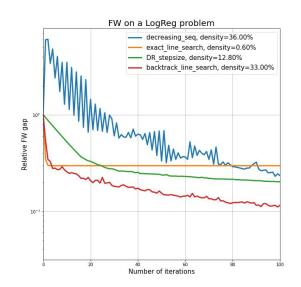


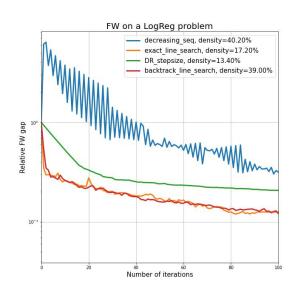
## Synthetic dataset MSE:

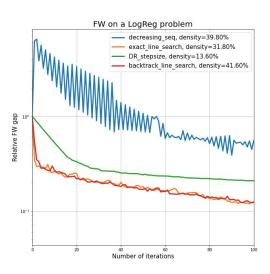




## **Modelon dataset logreg:**







# Thank you for your attention!