# Proximal Gradient Methods with Adaptive Subspace Sampling

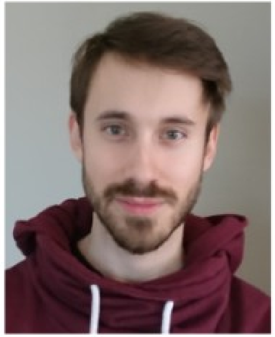
***Dmitry Grishchenko***

MOTOR 2021
Online

5 July 2021

# Collaborators

F. Iutzeler
LJK



J. Malick
CNRS, LJK

**Dmitry Grishchenko, Franck Iutzeler, and Jérôme Malick.** *Proximal gradient methods with adaptive subspace sampling.* Mathematics of Operations Research, 2021.

# Outline

Motivation

Randomized Subspace Descent

Adaptive Randomized Subspace Descent

# Outline

## Motivation

## Randomized Subspace Descent

## Adaptive Randomized Subspace Descent

# ML as an Optimization Problem

## Empirical Risk Minimization

Loss function: represents the difference between two arguments.

$$\underbrace{\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \ell(b_i, h(a_i, x))}_{f(x)}$$

# ML as an Optimization Problem

## Empirical Risk Minimization

Loss function: represents the difference between two arguments.

$$\underbrace{\min_{x\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m} \ell(b_i, h(a_i, x))}_{f(x)}$$

**Learning is a compromise between accuracy and complexity**

# ML as an Optimization Problem

## Structural Risk Minimization

Loss function: represents the difference between two arguments.

$$\min_{x \in \mathbb{R}^n} \underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(b_i, h(a_i, x))}_{f(x)} + r(x)$$

Regularization penalty.

# ML as an Optimization Problem

## Structural Risk Minimization

Smooth, convex.

$$\min_{x \in \mathbb{R}^n} \underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(b_i, h(a_i, x))}_{f(x)} + r(x)$$

Convex, non-smooth.

# ML as an Optimization Problem

## Structural Risk Minimization

Smooth, convex.

$$\min_{x\in\mathbb{R}^n} \underbrace{\frac{1}{m}\sum_{i=1}^{m}\ell(b_i, h(a_i, x))}_{f(x)} + r(x)$$

Convex, non-smooth.

## Why non smoothness?

# ML as an Optimization Problem

## Structural Risk Minimization

Smooth, convex.

$$\min_{x\in\mathbb{R}^n} \underbrace{\frac{1}{m}\sum_{i=1}^{m}\ell(b_i, h(a_i, x))}_{f(x)} + r(x)$$

Convex, non-smooth.

## Why non smoothness?

**To enforce some structure of the optimal solution.**

**Sparse solution** $r = \|\cdot\|_1,$

**e.g. feature selection problems**

**Fixed variation** $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|.$

**e.g. signal processing**

**Samuel Vaiter et al.** *Model selection with low complexity priors.* Information and Inference: A Journal of the IMA 4.3 (2015): 230-287.

# Proximal Gradient Descent

Let us consider a composite optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) + r(x),$$

where $f$ is $L$-smooth and convex, and $r$ is convex, l.s.c.

# Proximal Gradient Descent

Let us consider a composite optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) + r(x),$$

where $f$ is $L$-smooth and convex, and $r$ is convex, l.s.c.

## Proximal operator

$$\mathbf{prox}_r(y) = \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ r(x) + \frac{1}{2}\|x - y\|_2^2 \right\}.$$

This operator is well defined for convex $r$ and has a closed form solution for relatively simple $r$.

# Proximal Gradient Descent

Let us consider a composite optimization problem

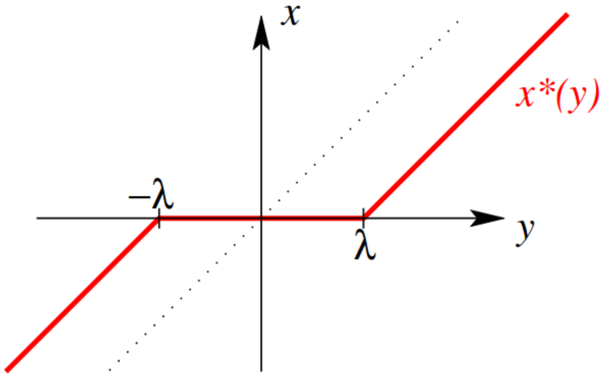
$$\min_{x \in \mathbb{R}^n} f(x) + r(x),$$

where $f$ is $L$-smooth and convex, and $r$ is convex, l.s.c.

## Proximal operator

$$\mathbf{prox}_r(y) = \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ r(x) + \frac{1}{2}\|x - y\|_2^2 \right\}.$$

$r = \lambda\|\cdot\|_1$




This operator is well defined for convex $r$ and has a closed form solution for relatively simple $r$.



MOTOR 2021

# Proximal Gradient Descent

Let us consider a composite optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) + r(x),$$

where $f$ is $L$-smooth and convex, and $r$ is convex, l.s.c.

## Proximal gradient descent

**Step 1** $y^k = x^k - \gamma \nabla f(x)$ **forward (gradient) step.**

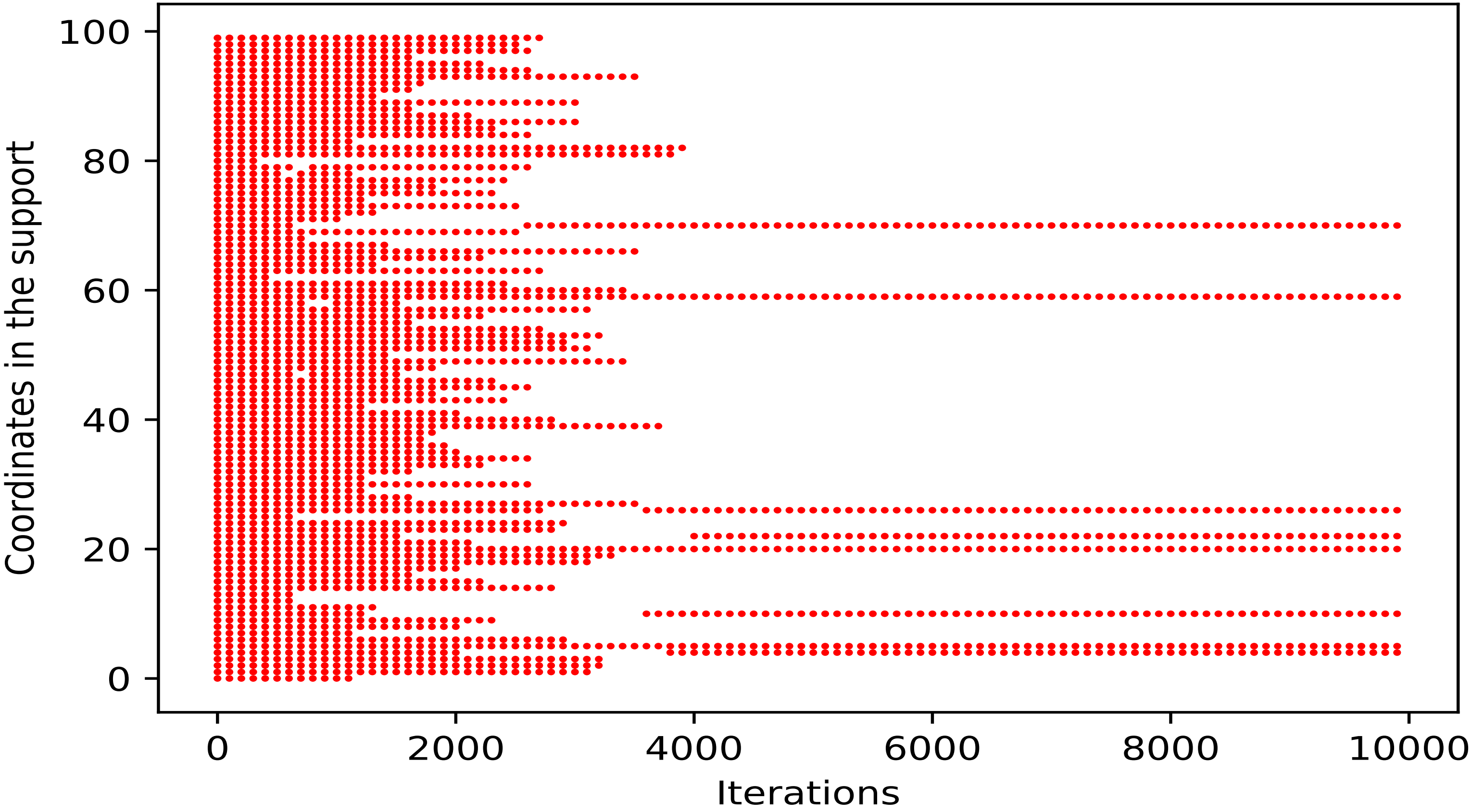
**Step 2** $x^{k+1} = \mathbf{prox}_{\gamma r}(y^k)$ **backward (proximal) step.**

**R Tyrrell Rockafellar.** *Monotone operators and the proximal point algorithm.* SIAM journal on control and optimization, 14(5):877–898, 1976.

# Identification

## One nice thing

**Proximal methods identify a near optimal subspace.**

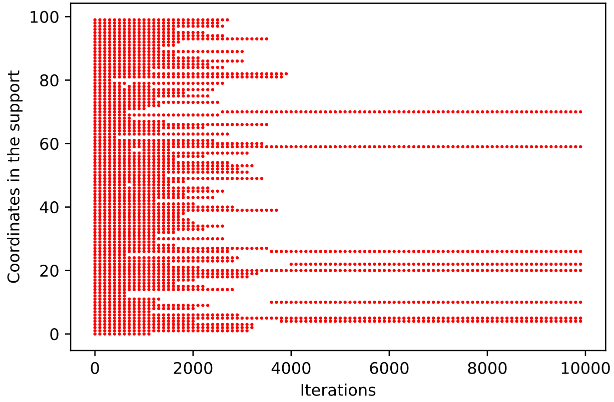



Synthetic LASSO problem $\min \frac{1}{2}\|Ax-b\|_2^2 + \lambda_1\|x\|_1$ for random generated matrix $A \in \mathbb{R}^{100\times 100}$ and vector $b \in \mathbb{R}^{100}$ and hyperparameter $\lambda_1$ chosen to reach 8% of density (amount of non-zero coordinates) of the final solution.

# Identification

## One nice thing

### Proximal methods identify a near optimal subspace.




### Sparsity vector

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ be a family of subspaces of $\mathbb{R}^n$ with $m$ elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$(\mathsf{S}_{\mathcal{M}}(x))_{[i]} = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and } 1 \text{ elsewhere.}$$

# Identification

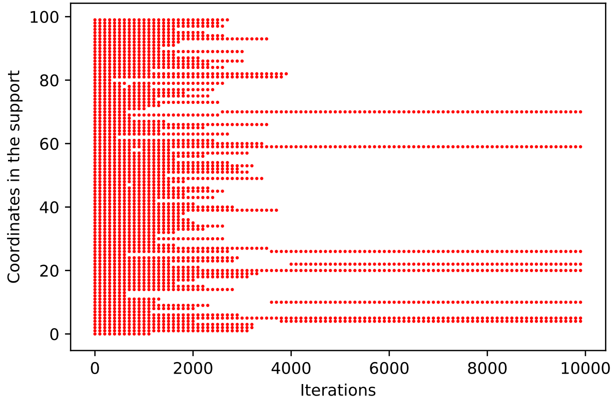
## One nice thing

### Proximal methods identify a near optimal subspace.

### Sparsity vector

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ be a family of subspaces of $\mathbb{R}^n$ with $m$ elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$(\mathsf{S}_{\mathcal{M}}(x))_{[i]} = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and } 1 \text{ elsewhere.}$$




The collection $\mathcal{M} = \{\mathcal{M}_i\}_{1 \leq i \leq n}$ is the set of subspaces $\mathcal{M}_i$ with $\text{supp}(x) = [n] \setminus \{i\}$ for all $x \in \mathcal{M}_i$.

# Identification

$$x^\star = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x) + r(x)$$

HUAWEI

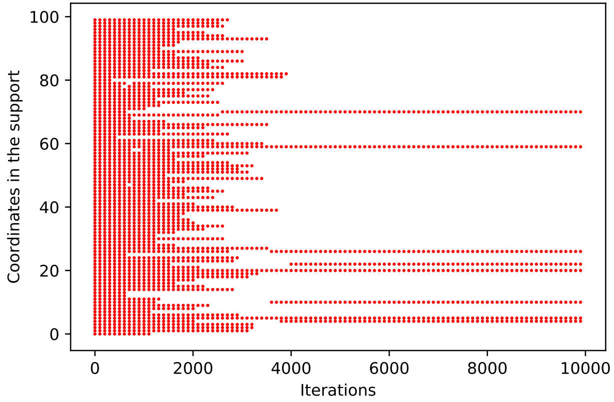
## One nice thing

**Proximal methods identify a near optimal subspace.**

### Sparsity vector

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ be a family of subspaces of $\mathbb{R}^n$ with $m$ elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$(\mathsf{S}_{\mathcal{M}}(x))_{[i]} = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and } 1 \text{ elsewhere.}$$




The collection $\mathcal{M} = \{\mathcal{M}_i\}_{1\leq i\leq n}$ is the set of subspaces $\mathcal{M}_i$ with $\operatorname{supp}(x) = [n] \setminus \{i\}$ for all $x \in \mathcal{M}_i$.

### Theorem (Enlarged identification)

Let $(u^k)$ be an $\mathbb{R}^n$-valued sequence converging almost surely to $u^\star$ and define sequence $(x^k)$ as $x^k = \mathbf{prox}_{\gamma r}(u^k)$ and $x^\star = \mathbf{prox}_{\gamma r}(u^\star)$. Then $(x^k)$ identifies some subspaces with probability one; more precisely for any $\varepsilon > 0$, with probability one, after some finite time,

$$\mathsf{S}_{\mathcal{M}}(x^\star) \ \leq\ \mathsf{S}_{\mathcal{M}}(x^k) \ \leq\ \max\left\{\mathsf{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma r}(u)) \colon u \in \mathcal{B}(u^\star, \varepsilon)\right\}.$$


**Franck Iutzeler and Jérôme Malick.** *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications.* Set-Valued and Variational Analysis (2020): 1-18.

# Identification

$$x^\star = \operatorname*{argmin}_{x\in\mathbb{R}^n} f(x) + r(x)$$

## One nice thing

**Proximal methods identify a near optimal subspace.**

### Sparsity vector

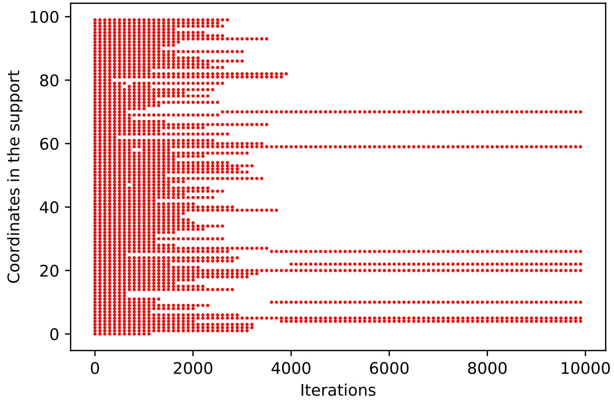
Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ be a family of subspaces of $\mathbb{R}^n$ with $m$ elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$(\mathsf{S}_{\mathcal{M}}(x))_{[i]} = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and 1 elsewhere.}$$

### Theorem (Enlarged identification)

Let $(u^k)$ be an $\mathbb{R}^n$-valued sequence converging almost surely to $u^\star$ and define sequence $(x^k)$ as $x^k = \mathbf{prox}_{\gamma r}(u^k)$ and $x^\star = \mathbf{prox}_{\gamma r}(u^\star)$. Then $(x^k)$ identifies some subspaces with probability one; more precisely for any $\varepsilon > 0$, with probability one, after some finite time,

$$\mathsf{S}_{\mathcal{M}}(x^\star) \;\leq\; \mathsf{S}_{\mathcal{M}}(x^k) \;\leq\; \max\left\{\mathsf{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma r}(u)) \colon u \in \mathcal{B}(u^\star, \varepsilon)\right\}.$$




The collection $\mathcal{M} = \{\mathcal{M}_i\}_{1\leq i\leq n}$ is the set of subspaces $\mathcal{M}_i$ with $\operatorname{supp}(x) = [n] \setminus \{i\}$ for all $x \in \mathcal{M}_i$.

$$\begin{aligned}\operatorname{supp}(x^\star) &\subseteq \operatorname{supp}(x^k)\\ &\subseteq \max_{u\in\mathcal{B}(u^\star,\varepsilon)} \{\operatorname{supp}(\mathbf{prox}_{\gamma r}(u))\}.\end{aligned}$$


**Franck Iutzeler and Jérôme Malick.** *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications.* Set-Valued and Variational Analysis (2020): 1-18.

# Identification

$$x^\star = \operatorname*{argmin}_{x\in\mathbb{R}^n} f(x) + r(x)$$

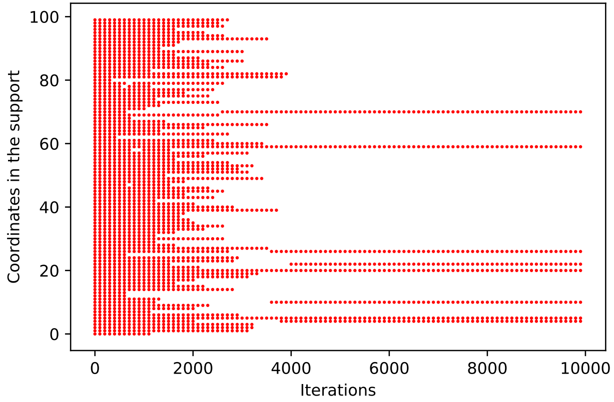
## One nice thing

**Proximal methods identify a near optimal subspace.**

**Sparsity vector**

Let $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_m\}$ be a family of subspaces of $\mathbb{R}^n$ with $m$ elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$(\mathsf{S}_{\mathcal{M}}(x))_{[i]} = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and } 1 \text{ elsewhere.}$$




The collection $\mathcal{M} = \{\mathcal{M}_i\}_{1\le i\le n}$ is the set of subspaces $\mathcal{M}_i$ with $\operatorname{supp}(x) = [n] \setminus \{i\}$ for all $x \in \mathcal{M}_i$.

### Theorem (Enlarged identification)

Let $(u^k)$ be an $\mathbb{R}^n$-valued sequence converging almost surely to $u^\star$ and define sequence $(x^k)$ as $x^k = \mathbf{prox}_{\gamma r}(u^k)$ and $x^\star = \mathbf{prox}_{\gamma r}(u^\star)$. Then $(x^k)$ identifies some subspaces with probability one; more precisely for any $\varepsilon > 0$, with probability one, after some finite time,

$$\mathsf{S}_{\mathcal{M}}(x^\star) \;\le\; \mathsf{S}_{\mathcal{M}}(x^k) \;\le\; \max\left\{\mathsf{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma r}(u)) : u \in \mathcal{B}(u^\star, \varepsilon)\right\}.$$

$$\operatorname{supp}(x^\star) \subseteq \operatorname{supp}(x^k)$$
$$\subseteq \max_{u\in\mathcal{B}(u^\star,\varepsilon)} \{\operatorname{supp}(\mathbf{prox}_{\gamma r}(u))\}.$$

**The same = verifies QC.**


**Franck Iutzeler and Jérôme Malick.** *Nonsmoothness in Machine Learning: specific structure, proximal identification, and applications.* Set-Valued and Variational Analysis (2020): 1-18.

# Outline

Motivation

**Randomized Subspace Descent**

Adaptive Randomized Subspace Descent

# Randomized Coordinate Descent

**Full gradient computation is expensive.**

# Randomized Coordinate Descent

**Full gradient computation is expensive.**

**Coordinate descent methods is a class of iterative methods in which only one coordinate (block) is updated on every iteration.**

# Randomized Coordinate Descent

**Full gradient computation is expensive.**

**Coordinate descent methods is a class of iterative methods in which only one coordinate (block) is updated on every iteration.**

**Example 1 (smooth).**

$$x^{k+1} = x^k - \gamma \nabla f(x)_{[i^k]}$$

# Randomized Coordinate Descent

**Full gradient computation is expensive.**

**Coordinate descent methods is a class of iterative methods in which only one coordinate (block) is updated on every iteration.**

**Example 1 (smooth).**

$$x^{k+1} = x^k - \gamma \nabla f(x)_{[i^k]}$$

**Example 2 (separable regularizer).**

$$r(x) = \sum_{i=1}^{n} r_i(x_{[i]}) \;\Rightarrow\; \mathbf{prox}_{\gamma r}(x)_{[i]} = \mathbf{prox}_{\gamma r_i}(x_{[i]}).$$

$$\mathrm{x}_{[i^k]}^{k+1} \leftarrow \mathbf{prox}_{\gamma r_{i^k}}\left(x_{[i^k]}^k - \gamma \nabla_{[i^k]} f(x^k)\right)$$

# Randomized Coordinate Descent

**Full gradient computation is expensive.**

**Coordinate descent methods is a class of iterative methods in which only one coordinate (block) is updated on every iteration.**

**Example 1 (smooth).**

$$x^{k+1} = x^k - \gamma \nabla f(x)_{[i^k]}$$

**Example 2 (separable regularizer).**

$$r(x) = \sum_{i=1}^{n} r_i(x_{[i]}) \;\Rightarrow\; \mathbf{prox}_{\gamma r}(x)_{[i]} = \mathbf{prox}_{\gamma r_i}(x_{[i]}).$$

$$\mathrm{x}_{[i^k]}^{k+1} \leftarrow \mathbf{prox}_{\gamma r_{i^k}}\left(x_{[i^k]}^{k} - \gamma \nabla_{[i^k]} f(x^k)\right)$$

**Drawback: explicit use of the separability of the regularizer.**

**Peter Richtárik and Martin Takáč.** *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function.* Mathematical Programming 144.1-2 (2014): 1-38.

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

**Olivier Fercoq and Pascal Bianchi**. *A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions.* SIAM Journal on Optimization 29.1 (2019): 100-134.

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x_{[i^k]}^{k+1} \leftarrow \mathbf{prox}_{\gamma r_{i^k}} \left( x_{[i^k]}^k - \gamma \nabla_{[i^k]} f(x^k) \right)$$

# Randomized Subspace Descent

## What if the regularizer is not separable?

e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$\cancel{x_{[i^k]}^{k+1} \leftarrow \mathbf{prox}_{\gamma r_{i^k}}\left(x_{[i^k]}^k - \gamma \nabla_{[i^k]} f(x^k)\right)}$$

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|.$

$$x^{k+1} = \mathbf{prox}_{\gamma r_{i^k}}\left(x^k_{[i^k]} - \gamma \nabla_{[i^k]} f(x^k)\right) + \left[x^k\right]_{\bar{i}^k}$$

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r}\left(\left[y^k\right]_{i^k} + \left[y^{k-1}\right]_{\bar{i}^k}\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r}\left(\left[y^k\right]_{i^k} + \left[y^{k-1}\right]_{\bar{i}^k}\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

**In this reformulation the separability is not required!**

# Randomized Subspace Descent

## What if the regularizer is not separable?

e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r}\left(\left[y^k\right]_{i^k} + \left[y^{k-1}\right]_{\bar{i}^k}\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

Two orthogonal projections
onto orthogonal spaces!

## In this reformulation the separability is not required!



MOTOR 2021

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r} \left(P\left(y^k\right) + (I - P)\left(y^{k-1}\right)\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

**In this reformulation the separability is not required!**

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r}\left(P\left(y^k\right) + (I - P)\left(y^{k-1}\right)\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

**In this reformulation the separability is not required!**

**General orthogonal projections are used!**

# Randomized Subspace Descent

**What if the regularizer is not separable?** e.g. $r = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$.

$$x^{k+1} = \mathbf{prox}_{\gamma r}\left(P\left(y^k\right) + (I - P)\left(y^{k-1}\right)\right),$$

where $y^k = x^k - \gamma \nabla f(x^k)$.

**In this reformulation the separability is not required!**

**General orthogonal projections are used!**

**Does it work like this?**



MOTOR 2021

# Examples: Subspaces

# Examples: Subspaces

**Example 3.**

Let us consider the set of subspaces $\mathcal{C}_i$ such that $\mathcal{C}_i$ is $i$-th coordinate line. Select an orthogonal projection onto the $\mathcal{C}_i$ with probability $\frac{1}{n-1}$ $\forall i \in [2, n]$ and 0 for the 1-st.

# Examples: Subspaces

**Example 3.**

Let us consider the set of subspaces $\mathcal{C}_i$ such that $\mathcal{C}_i$ is $i$-th coordinate line. Select an orthogonal projection onto the $\mathcal{C}_i$ with probability $\frac{1}{n-1}$ $\forall i \in [2, n]$ and $0$ for the 1-st.

**Does not work if the first coordinates of the starting and the optimal point are different.**

# Examples: Subspaces

**Example 3.**

Let us consider the set of subspaces $\mathcal{C}_i$ such that $\mathcal{C}_i$ is $i$-th coordinate line. Select an orthogonal projection onto the $\mathcal{C}_i$ with probability $\frac{1}{n-1}$ $\forall i \in [2, n]$ and 0 for the 1-st.

**Does not work if the first coordinates of the starting and the optimal point are different.**

**Covering family of subspaces**

Let $\mathcal{C} = \{\mathcal{C}_i\}_i$ be a family of subspaces of $\mathbb{R}^n$. We say that $\mathcal{C}$ is *covering* if it spans the whole space, i.e. if $\sum_i \mathcal{C}_i = \mathbb{R}^n$.

# Admissible Selection

# Admissible Selection

Let $\mathcal{C}$ be a covering family of subspaces of $\mathbb{R}^n$. A selection $\mathfrak{S}$ is defined from the set of all subsets of $\mathcal{C}$ to the set of the subspaces of $\mathbb{R}^n$ as

$$\mathfrak{S}(\omega) = \sum_{j=1}^{s} \mathcal{C}_{i_j} \qquad \text{for } \omega = \{\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_s}\}.$$

The selection $\mathfrak{S}$ is *admissible* if $\mathbb{P}[x \in \mathfrak{S}^\perp] < 1$ for all $x \in \mathbb{R}^n \setminus \{0\}$.

# Admissible Selection

Let $\mathcal{C}$ be a covering family of subspaces of $\mathbb{R}^n$. A selection $\mathfrak{S}$ is defined from the set of all subsets of $\mathcal{C}$ to the set of the subspaces of $\mathbb{R}^n$ as

$$\mathfrak{S}(\omega) = \sum_{j=1}^{s} \mathcal{C}_{i_j} \qquad \text{for } \omega = \{\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_s}\}.$$

The selection $\mathfrak{S}$ is *admissible* if $\mathbb{P}[x \in \mathfrak{S}^\perp] < 1$ for all $x \in \mathbb{R}^n \setminus \{0\}$.

If a selection $\mathfrak{S}$ is admissible then $\mathsf{P} := \mathbb{E}[P_\mathfrak{S}]$ is a positive definite matrix.

In this case, we denote by $\lambda_{\min}(\mathsf{P}) > 0$ and $\lambda_{\max}(\mathsf{P}) \leq 1$ its minimal and maximal eigenvalues.

# Algorithm 1: RPSD

**Algorithm 1** Randomized Proximal Subspace Descent - `RPSD`

Input: $\mathsf{Q} = \mathsf{P}^{-\frac{1}{2}}$
Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma r}(\mathsf{Q}^{-1}(z^0))$
**for** $k = 1, \ldots$ **do**
    $y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right)$
    $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + (I - P_{\mathfrak{S}^k})\left(z^{k-1}\right)$
    $x^{k+1} = \mathbf{prox}_{\gamma r}\left(\mathsf{Q}^{-1}\left(z^k\right)\right)$
**end for**

# Algorithm 1: RPSD

**Algorithm 1** Randomized Proximal Subspace Descent - `RPSD`

1: Input: $\mathsf{Q} = \mathsf{P}^{-\frac{1}{2}}$
2: Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma r}(\mathsf{Q}^{-1}(z^0))$
3: **for** $k = 1, \ldots$ **do**
4: $\quad y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right)$
5: $\quad z^k = P_{\mathfrak{S}^k}\left(y^k\right) + (I - P_{\mathfrak{S}^k})\left(z^{k-1}\right)$ ← Project
6: $\quad x^{k+1} = \mathbf{prox}_{\gamma r}\left(\mathsf{Q}^{-1}\left(z^k\right)\right)$
7: **end for**

``Sketch''

# RPSD: Convergence Result

**Assumption (on randomness)**

Given a covering family $\mathcal{C} = \{\mathcal{C}_i\}$ of subspaces, we consider a sequence $\mathfrak{S}^1, \mathfrak{S}^2, .., \mathfrak{S}^k$ of admissible selections, which is i.i.d.

# RPSD: Convergence Result

**Assumption (on randomness)**

Given a covering family $\mathcal{C} = \{\mathcal{C}_i\}$ of subspaces, we consider a sequence $\mathfrak{S}^1, \mathfrak{S}^2, .., \mathfrak{S}^k$ of admissible selections, which is i.i.d.

**Theorem (Convergence of RPSD)**

For any $\gamma \in (0, 2/(\mu+L)]$, the sequence $(x^k)$ of the iterates of $\texttt{RPSD}$ converges almost surely to the minimizer $x^\star$ with rate

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu + L}\right)^k C,$$

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2$.

# RPSD: Convergence Result

Consider the set of subspaces $\mathcal{C}_i$ such that $\mathcal{C}_i$ is $i$-th coordinate line. Consider the selection $\mathfrak{S}$ such that $\mathbb{P}\left[\mathcal{C}_i \in \mathfrak{S}\right] = p_i > 0$, then $\lambda_{\min}(\mathsf{P}) = \min_i p_i > 0$.

**Theorem (Convergence of RPSD)**

For any $\gamma \in (0, 2/(\mu+L)]$, the sequence $(x^k)$ of the iterates of `RPSD` converges almost surely to the minimizer $x^\star$ with rate

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu + L}\right)^k C,$$

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2$.

# RPSD: Proof Sketch

**Lemma 1**

From the minimizer $x^\star$, define the fixed points $z^\star = y^\star = \mathsf{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$ of the sequences $(y^k)$ and $(z^k)$. Then

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] = \|z^{k-1} - z^\star\|_2^2 + \|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2,$$

where $\mathcal{F}^k = \sigma(\{\mathfrak{S}_\ell\}_{\ell \leq k})$ is the filtration of the past random subspaces.

$$z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$$

# RPSD: Proof Sketch

**Lemma 1**

From the minimizer $x^\star$, define the fixed points $z^\star = y^\star = \mathsf{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$ of the sequences $(y^k)$ and $(z^k)$. Then

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] = \|z^{k-1} - z^\star\|_2^2 + \|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2,$$

where $\mathcal{F}^k = \sigma(\{\mathfrak{S}_\ell\}_{\ell \leq k})$ is the filtration of the past random subspaces.

**Lemma 2**

Using the same notations as in Lemma 1

$$\|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2 \leq -\lambda_{\min}(\mathsf{P}) \frac{2\gamma\mu L}{\mu + L} \|z^{k-1} - z^\star\|_2^2.$$

# RPSD: Proof Sketch

HUAWEI

**Lemma 1**

From the minimizer $x^\star$, define the fixed points $z^\star = y^\star = \mathsf{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$ of the sequences $(y^k)$ and $(z^k)$. Then

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] = \|z^{k-1} - z^\star\|_2^2 + \|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2,$$

where $\mathcal{F}^k = \sigma(\{\mathfrak{S}_\ell\}_{\ell \leq k})$ is the filtration of the past random subspaces.

Identification!

**Lemma 2**

Using the same notations as in Lemma 1

$$\|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2 \leq -\lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu + L}\|z^{k-1} - z^\star\|_2^2.$$



MOTOR 2021

# Examples: TV Projections

$$r = \lambda \sum_{i=1}^{n-1} \left| x_{[i]} - x_{[i+1]} \right|$$

**Fixed variation sparsity = small amount of blocks of equal coordinates.**

# Examples: TV Projections

$$r = \lambda \sum_{i=1}^{n-1} \left| x_{[i]} - x_{[i+1]} \right|$$

**Fixed variation sparsity = small amount of blocks of equal coordinates.**

## Projection on such set

$$P_{\mathfrak{S}} = \begin{array}{c} \begin{array}{cccccccc} \multicolumn{3}{c}{\overbrace{\qquad\qquad\qquad}^{n_1}} & \qquad & \qquad & \multicolumn{3}{c}{\overbrace{\qquad\qquad\qquad}^{n-n_s}} \end{array} \\ \begin{pmatrix} \frac{1}{n_1} & \cdots & \frac{1}{n_1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{1}{n_1} & \cdots & \frac{1}{n_1} & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 & \frac{1}{n-n_s} & \cdots & \frac{1}{n-n_s} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{1}{n-n_s} & \cdots & \frac{1}{n-n_s} \end{pmatrix} \end{array} \begin{array}{l} \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} n_1 \\ \vphantom{\begin{matrix}a\\a\end{matrix}} \\ \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} n-n_s \end{array}$$

# Outline

## Motivation

## Randomized Subspace Descent

# Adaptive Randomized Subspace Descent

# Algorithm 2: ARPSD

**Algorithm 2** Adaptive Randomized Proximal Subspace Descent - `ARPSD`

Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.
**for** $k = 1, \ldots$ **do**
  $y^k = \mathsf{Q}_\ell \left(x^k - \gamma \nabla f\left(x^k\right)\right)$
  $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
  $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}_\ell^{-1}\left(z^k\right)\right)$
  **if** an adaptation is decided **then**
    $\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$
    Generate a new admissible selection
    Compute $\mathsf{Q}_\ell = \mathsf{P}_\ell^{-\frac{1}{2}}$ and $\mathsf{Q}_\ell^{-1}$
    Rescale $z^k \leftarrow \mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1} z^k$
  **end if**
**end for**

# Algorithm 2: ARPSD

**Algorithm 2** Adaptive Randomized Proximal Subspace Descent - `ARPSD`

Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.
**for** $k = 1, \ldots$ **do**
  $y^k = \mathsf{Q}_\ell \left(x^k - \gamma \nabla f\left(x^k\right)\right)$
  $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
  $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}_\ell^{-1}\left(z^k\right)\right)$
  **if** an adaptation is decided **then**
    $\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$
    Generate a new admissible selection
    Compute $\mathsf{Q}_\ell = \mathsf{P}_\ell^{-\frac{1}{2}}$ and $\mathsf{Q}_\ell^{-1}$
    Rescale $z^k \leftarrow \mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1} z^k$
  **end if**
**end for**

# Algorithm 2: ARPSD

**Algorithm 2** Adaptive Randomized Proximal Subspace Descent - ARPSD

Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.
**for** $k = 1, \ldots$ **do**
  $y^k = \mathsf{Q}_\ell \left(x^k - \gamma \nabla f\left(x^k\right)\right)$
  $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
  $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}_\ell^{-1}\left(z^k\right)\right)$
  **if** an adaptation is decided **then**
    $\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$
    Generate a new admissible selection
    Compute $\mathsf{Q}_\ell = \mathsf{P}_\ell^{-\frac{1}{2}}$ and $\mathsf{Q}_\ell^{-1}$
    Rescale $z^k \leftarrow \mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1} z^k$
  **end if**
**end for**

$$\begin{cases} (\mathcal{C}_i \bigcap \mathcal{M}_i) \subseteq \bigcap_j \mathcal{C}_j \\ \mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n \end{cases}$$

$$\mathbb{P}[\mathcal{C}_i \in \mathfrak{S}^{k+1}] = \begin{cases} p & \text{if } x^{k+1} \in \mathcal{M}_i \Leftrightarrow [\mathsf{S}_\mathcal{M}(x^{k+1})]_i = 0 \\ 1 & \text{elsewhere} \end{cases}$$

# Algorithm 2: ARPSD

HUAWEI

**Algorithm 2** Adaptive Randomized Proximal Subspace Descent - `ARPSD`

Initialize $z^0$, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.

**for** $k = 1, \ldots$ **do**

$y^k = \mathsf{Q}_\ell\left(x^k - \gamma \nabla f\left(x^k\right)\right)$

$z^k = P_{\mathfrak{S}^k}\left(y^k\right) + (I - P_{\mathfrak{S}^k})\left(z^{k-1}\right)$

$x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}_\ell^{-1}\left(z^k\right)\right)$

**if** an adaptation is decided **then**

$\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$

Generate a new admissible selection

Compute $\mathsf{Q}_\ell = \mathsf{P}_\ell^{-\frac{1}{2}}$ and $\mathsf{Q}_\ell^{-1}$

Rescale $z^k \leftarrow \mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1} z^k$

**end if**

**end for**



MOTOR 2021

# Adaptation Process

Let us specify ARPSD with the following simple adaptation strategy. We take a fixed upper bound on the adaptation cost and a fixed lower bound on uniformity:

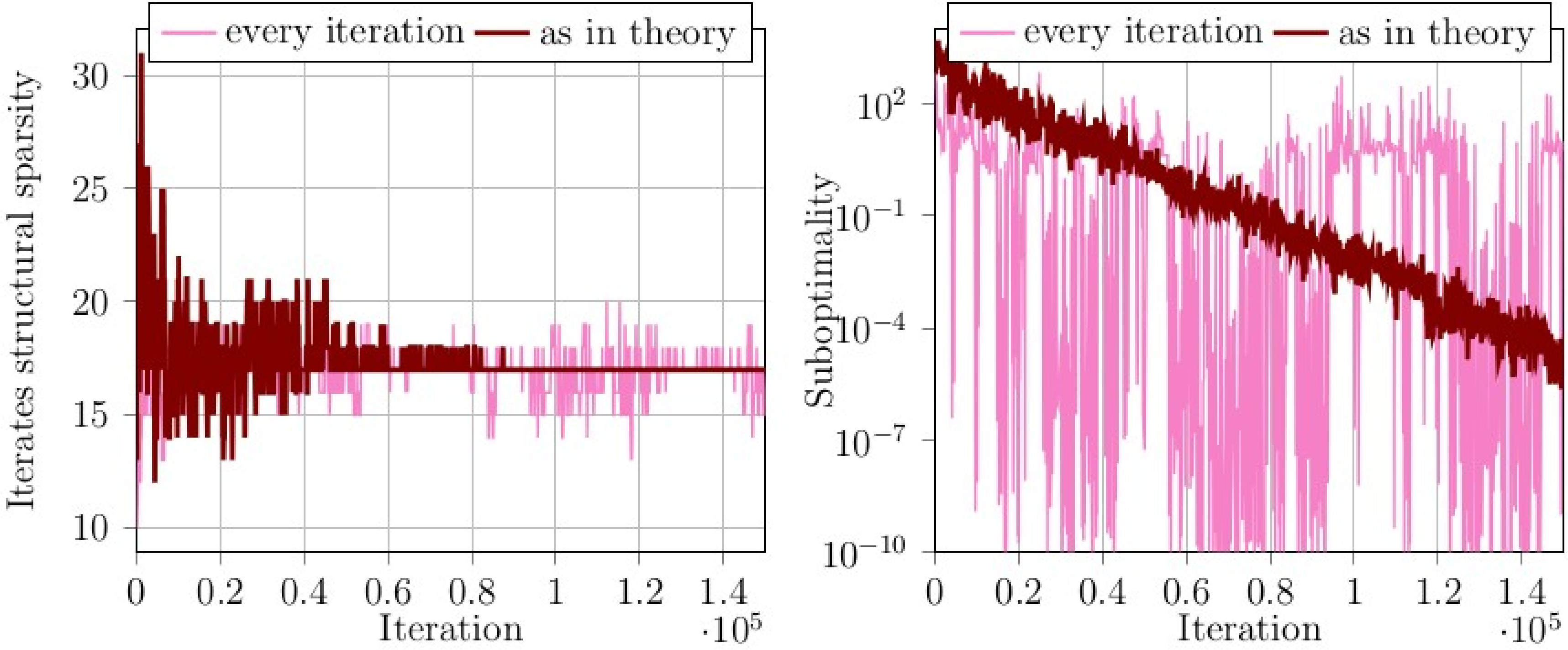
$$\|\mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1}\|_2^2 \leq \mathbf{a} \qquad \lambda_{\min}(\mathsf{P}_\ell) \geq \lambda.$$

Then from the rate $1-\alpha = 1 - 2\gamma\mu L\lambda/(\mu+L)$, we can perform an adaptation every

$$\mathbf{c} = \lceil \log(\mathbf{a})/\log\left((2-\alpha)/(2-2\alpha)\right)\rceil$$

iterations, so that $\mathbf{a}(1-\alpha)^{\mathbf{c}} = (1-\alpha/2)^{\mathbf{c}}$ and $k_\ell = \ell\mathbf{c}$.

# Adaptation Process



# ARPSD: Convergence Result

# ARPSD: Convergence Result

**Assumption (on randomness)**

For all $k > 0$, $\mathfrak{S}^k$ is $\mathcal{F}^k$-measurable and admissible. Furthermore, if $k \notin \mathsf{L}$, $(\mathfrak{S}^k)$ is independent and identically distributed on $[k_\ell, k]$. The decision to adapt or not at time $k$ is $\mathcal{F}^k$-measurable, i.e. $(k_\ell)_\ell$ is a sequence of $\mathcal{F}^k$-stopping times.

# ARPSD: Convergence Result

**Assumption (on randomness)**

For all $k > 0$, $\mathfrak{S}^k$ is $\mathcal{F}^k$-measurable and admissible. Furthermore, if $k \notin \mathsf{L}$, $(\mathfrak{S}^k)$ is independent and identically distributed on $[k_\ell, k]$. The decision to adapt or not at time $k$ is $\mathcal{F}^k$-measurable, i.e. $(k_\ell)_\ell$ is a sequence of $\mathcal{F}^k$-stopping times.

**Theorem (Convergence of ARPSD)**

For any $\gamma \in (0, 2/(\mu + L)]$, the sequence $(x^k)$ of the iterates of `ARPSD` converges almost surely to the minimizer $x^\star$ with rate

$$\mathbb{E}\left[\|x^{k+1} - x^\star_\ell\|_2^2\right] \leq \left(1 - \frac{\lambda}{2}\frac{2\gamma\mu L}{\mu + L}\right)^k C.$$

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2$.

# ARPSD: Convergence Result

**Assumption (on randomness)**

For all $k > 0$, $\mathfrak{S}^k$ is $\mathcal{F}^k$-measurable and admissible. Furthermore, if $k \notin \mathsf{L}$, $(\mathfrak{S}^k)$ is independent and identically distributed on $[k_\ell, k]$. The decision to adapt or not at time $k$ is $\mathcal{F}^k$-measurable, i.e. $(k_\ell)_\ell$ is a sequence of $\mathcal{F}^k$-stopping times.

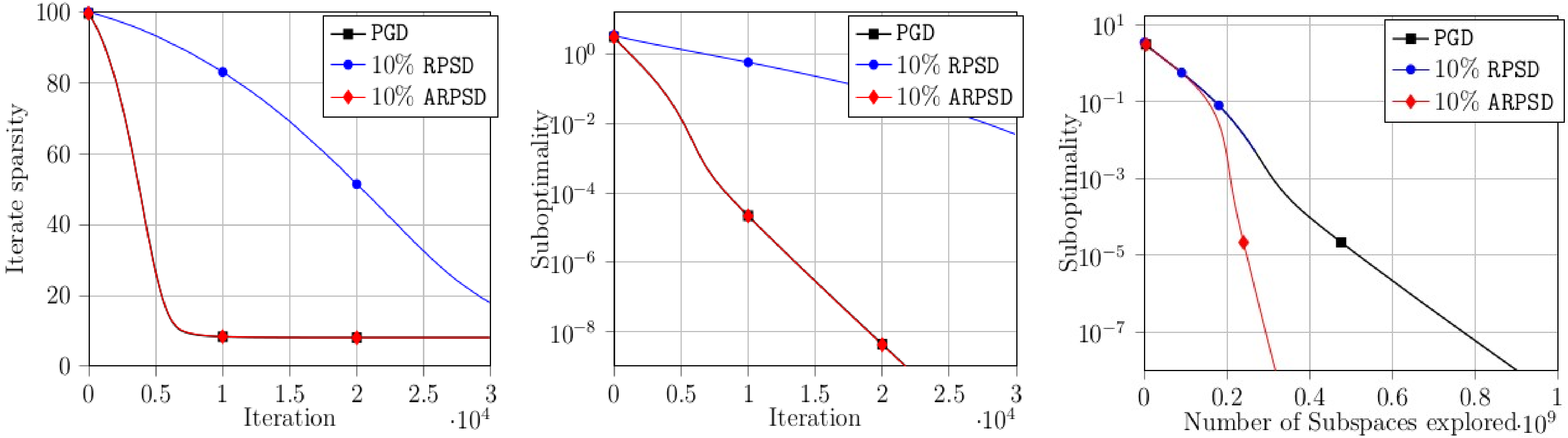
**Theorem (Convergence of ARPSD)**

For any $\gamma \in (0, 2/(\mu + L)]$, the sequence $(x^k)$ of the iterates of ARPSD converges almost surely to the minimizer $x^\star$ with rate

$$\mathbb{E}\left[\|x^{k+1} - x_\ell^\star\|_2^2\right] \leq \left(1 - \frac{\lambda}{2} \frac{2\gamma\mu L}{\mu + L}\right)^k C.$$

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2$.

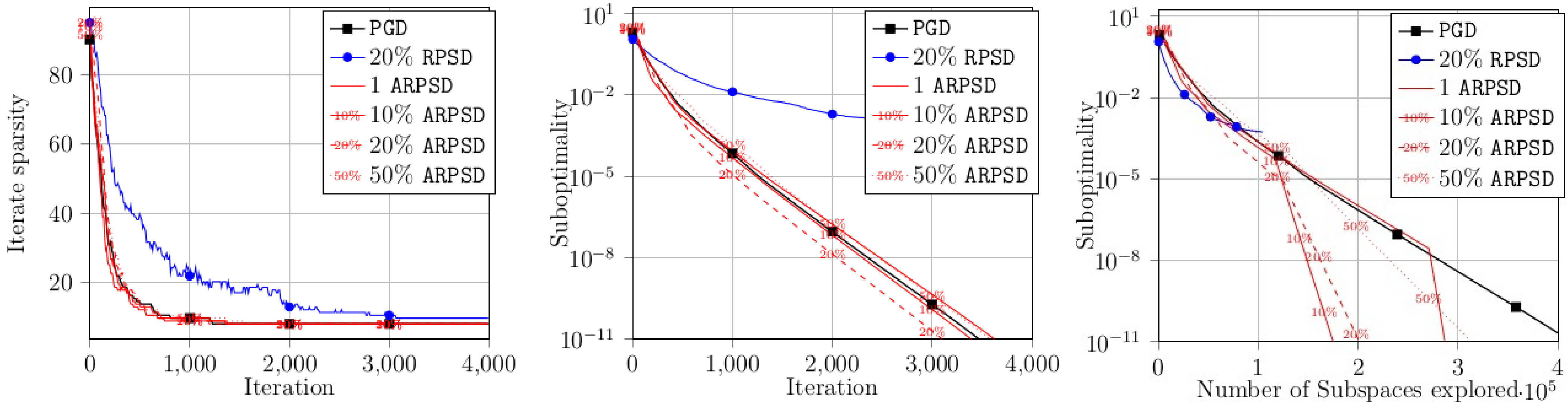
# Experiments: Inefficiency of RPSD




Logistic regression with elastic net regularizer on rcv1_train dataset ($n = 47236$ $m = 20242$).

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{m}\sum_{j=1}^{m}\log(1+\exp(-y_j z_j^\top x)) + \lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|_2^2$$

# Experiments: ARPSD with TV




1D-TV-regularized logistic regression on a1a dataset ($n = 123$ $m = 1605$).

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{m}\sum_{j=1}^{m}\log(1+\exp(-y_j z_j^\top x)) + \lambda_1 \sum_{i=1}^{n-1}|x_{[i]} - x_{[i+1]}| + \frac{\lambda_2}{2}\|x\|_2^2$$

# Strange Metric?

$$\min_{x\in\mathbb{R}^n} \underbrace{\frac{1}{m}\sum_{i=1}^{m}\ell(b_i, h(a_i, x))}_{f(x)} + r(x)$$

# Strange Metric?

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} \alpha_i \underbrace{\left[\frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i} \ell(b_j, h(a_j, x))\right]}_{f_i} + r(x),$$

where the full dataset $\mathcal{D}$ is split onto $M$ nonintersecting subsets $\mathcal{D}_i$ and $\alpha_i$ is the proportion of examples $\frac{|\mathcal{D}_i|}{m}$.

# Strange Metric?

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} \alpha_i \underbrace{\left[\frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i} \ell(b_j, h(a_j, x))\right]}_{f_i} + r(x),$$

where the full dataset $\mathcal{D}$ is split onto $M$ nonintersecting subsets $\mathcal{D}_i$ and $\alpha_i$ is the proportion of examples $\frac{|\mathcal{D}_i|}{m}$.

These subsets $\mathcal{D}_i$ can be split over machines.

# Strange Metric?

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} \alpha_i \underbrace{\left[\frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i} \ell(b_j, h(a_j, x))\right]}_{f_i} + r(x),$$

where the full dataset $\mathcal{D}$ is split onto $M$ nonintersecting subsets $\mathcal{D}_i$ and $\alpha_i$ is the proportion of examples $\frac{|\mathcal{D}_i|}{m}$.

These subsets $\mathcal{D}_i$ can be split over machines.

**Communication**

$$z^k = \sum_i \alpha_i z_i^k$$

**Master**

$$z_i^k = P_{\mathfrak{S}^k}\left(y_i^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z_i^{k-1}\right)$$

# Strange Metric?

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} \alpha_i \underbrace{\left[\frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i} \ell(b_j, h(a_j, x))\right]}_{f_i} + r(x),$$

where the full dataset $\mathcal{D}$ is split onto $M$ nonintersecting subsets $\mathcal{D}_i$ and $\alpha_i$ is the proportion of examples $\frac{|\mathcal{D}_i|}{m}$.

These subsets $\mathcal{D}_i$ can be split over machines.

**Bottleneck**

$$z^k = \sum_i \alpha_i z_i^k$$

**Master**

$$z_i^k = P_{\mathfrak{S}^k}\left(y_i^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z_i^{k-1}\right)$$

# Strange Metric?

$$\min_{x\in\mathbb{R}^n}\sum_{i=1}^{M}\alpha_i\underbrace{\left[\frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i}\ell(b_j,h(a_j,x))\right]}_{f_i}+r(x),$$

where the full dataset $\mathcal{D}$ is split onto $M$ nonintersecting subsets $\mathcal{D}_i$ and $\alpha_i$ is the proportion of examples $\frac{|\mathcal{D}_i|}{m}$.

These subsets $\mathcal{D}_i$ can be split over machines.

$$z^k = \sum_i \alpha_i z_i^k$$

**Bottleneck**

Amount of subspaces explored

**Master**

$$z_i^k = P_{\mathfrak{S}^k}\left(y_i^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z_i^{k-1}\right)$$


Thank You For
Your Attention!

# Practical for TV regularizer

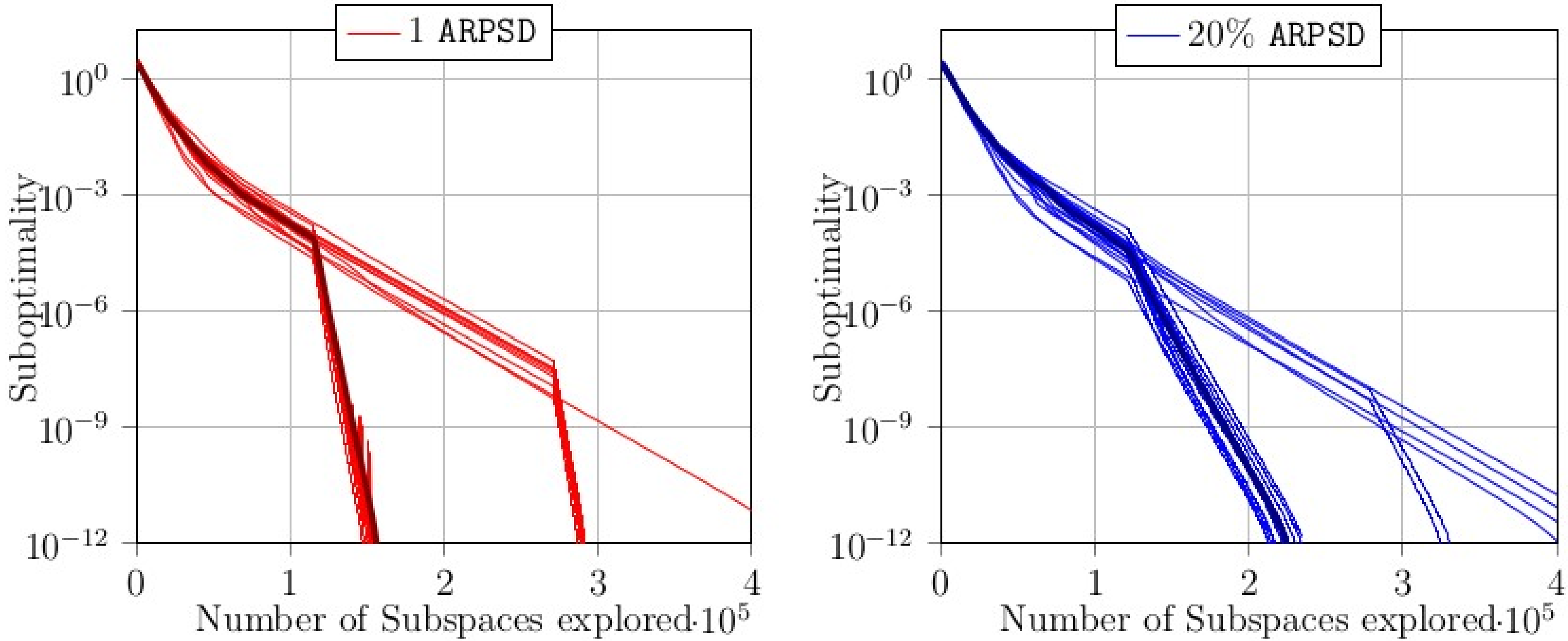
Consider the set of artificial jumps $\mathcal{S} = \{n_1, n_2, \ldots, n_{l-1}\}$ and denote by $\mathcal{R} = \{i \notin \mathcal{S} : [\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0\}$ the set of possible random entries. Fix the amount of sampled elements $s$ and sample "first" element $\mathcal{R}_0$ uniformly in $\mathcal{R} = \{\mathcal{R}_i\}_{1 \le i \le r}$. Select "first $s$" elements starting from $\mathcal{R}_f$ considering the cyclic structure of the list of elements $(\mathcal{R}_{r+1} = \mathcal{R}_1)$.

If $l$ is small enough, it will not change the sparsity property of the random projection $P_{\mathfrak{S}^k}$; however, this modification will force all the projections to be block-diagonal with blocks' ends on positions $n_1, \ldots n_{l-1}$. In contrast with jumps$(x^k)$ that we could not control, by adding $l$ artificial jumps, we could guarantee that each block of the $P_{\mathfrak{S}^k}$ has at most $\lceil n/l \rceil$ rows. Since every random projection has end of the block on positions $\{n_i\}_{1 \le i \le l-1}$. $\mathsf{P}_\ell$ also has such block structure and we could split the computation of $\mathsf{Q}_\ell^{-1}$ and $\mathsf{Q}_\ell$ into $l$ independent parts and could be done in parallel.

# Strategies for (A)RPSD

| | (non-adaptive) subspace descent RPSD | adaptive subspace descent ARPSD |
|---|---|---|
| Subspace family | $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$ | |
| Algorithm | $\begin{cases} y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right) \\ z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right) \\ x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}^{-1}\left(z^k\right)\right) \end{cases}$ | |
| Selection — Option 1 | $\mathcal{C}_i \in \mathfrak{S}^k$ with probability $p$ | $\mathcal{C}_i \in \mathfrak{S}^k$ with probability $\begin{cases} p & \text{if } x^k \in \mathcal{M}_i \Leftrightarrow [\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0 \\ 1 & \text{elsewhere} \end{cases}$ |
| Selection — Option 2 | Sample $s$ elements uniformly in $\mathcal{C}$ | Sample $s$ elements uniformly in $\{\mathcal{C}_i : x^k \in \mathcal{M}_i \text{ i.e. } [\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0\}$ and add *all* elements in $\{\mathcal{C}_j : x^k \notin \mathcal{M}_j \text{ i.e. } [\mathsf{S}_{\mathcal{M}}(x^k)]_j = 1\}$ |

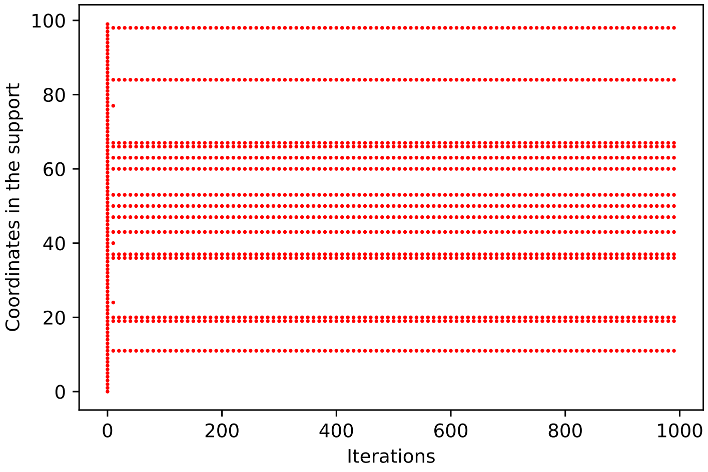
# Practical robustness

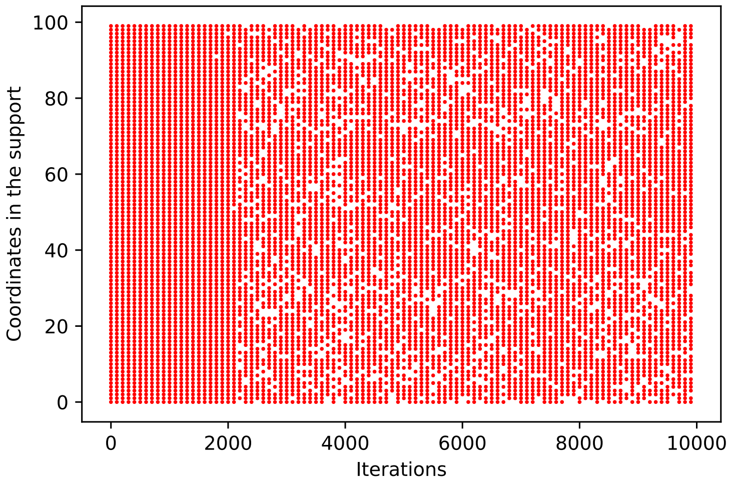



Logistic regression with elastic net regularizer on rcv1_train dataset ($n = 47236$ $m = 20242$).

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{m}\sum_{j=1}^{m}\log(1+\exp(-y_j z_j^\top x)) + \lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|_2^2$$

# Why not SGD




**Prox GD**

**Prox SGD (minibatch of size 10)**

Synthetic LASSO problem $\min \frac{1}{2}\|Ax - b\|_2^2 + \lambda_1 \|x\|_1$ for random generated matrix $A \in \mathbb{R}^{100\times 100}$ and vector $b \in \mathbb{R}^{100}$ and hyperparameter $\lambda_1$ chosen to reach 15% of density (amount of non-zero coordinates) of the final solution.

# Non-degeneracy

Another way to define the non-degeneracy for the problem

$$\min_{x \in \mathbb{R}^n} f(x) + r(x)$$

is the following:

$$\nabla f(x^\star) \in \operatorname{ri} \partial r(x^\star).$$

In case of $\ell_1$ regularizer $r(x) = \lambda_1 \|x\|_1$ this can be written explicitly as

$$\left| \nabla f(x^\star)_{[j]} \right| < \lambda_1 \qquad \text{for all } j \in \operatorname{supp}(x^\star).$$