A Levenberg-Marquardt method for large-scale noisy nonlinear least squares problems

Elisa Riccietti

Università degli Studi di Firenze
Dipartimento di Matematica e Informatica 'Ulisse Dini'

Joint work with: Stefania Bellavia (Università di Firenze), Serge Gratton (ENSEEIHT, Toulouse)

UNIVERSITÀ DEGLI STUDI FIRENZE

SIOPT 2017

Large scale problems with noisy function and noisy gradient

Let us consider the following nonlinear least squares problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2$$

where $F : \mathbb{R}^n \to \mathbb{R}^N$ with $N \geq n$, continuously differentiable.

Noisy function and noisy gradients

We are interested in **large scale** problems for which either:

- exact values for the function and the gradient are not available,
- computing exact values is computationally demanding.

Function approximations

- We relay on cheap approximations f_δ to f of known accuracy.
- We measure the accuracy of the approximations in x by

$$|f_\delta(x) - f(x)| \leq \delta, \quad \delta \text{ noise level.}$$

- We assume that the accuracy level can be improved along the optimization process.
- The approximation is updated through iterations: f_{δ_k}.

Jacobian and gradient approximation

- J_{δ_k} Jacobian matrix approximation,
- g_{δ_k} gradient approximation.

Typical applications

Machine learning, Data assimilation

Subsampling techniques

- Large set of data at disposal: $\{1, \dots, N\}$.
  Redundancy in the measurements $\rightarrow$ subsampling: $X_k \subseteq \{1, \dots, N\}$ such that $|X_k| = K_k \leq N$ is selected.
- $F_{\delta_k} : \mathbb{R}^n \rightarrow \mathbb{R}^{K_k}$ such that $(F_{\delta_k})_i = F_j$, $j \in X_k$ is built.
  If $X_k = \{2, 5, 7\}$ then $F_{\delta_k} = [F_2; F_5, F_7]^T$.

Typical applications

Machine learning, Data assimilation

Subsampling techniques

- Large set of data at disposal: $\{1, \dots, N\}$.
  Redundancy in the measurements $\rightarrow$ subsampling: $X_k \subseteq \{1, \dots, N\}$ such that $|X_k| = K_k \leq N$ is selected.
- $F_{\delta_k} : \mathbb{R}^n \rightarrow \mathbb{R}^{K_k}$ such that $(F_{\delta_k})_i = F_j$, $j \in X_k$ is built.
  If $X_k = \{2, 5, 7\}$ then $F_{\delta_k} = [F_2; F_5, F_7]^T$.
- $f_{\delta_k}(x) = \frac{1}{2}\|F_{\delta_k}(x)\|^2 \rightarrow$ can be improved considering more observations, i.e. increasing K_k.

We propose a **Levenberg-Marquardt method**.

Algorithm : k-th iteration

1. Step computation: define the LM model

$$\min_{p \in \mathbb{R}^n} m_k(x_k + p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

   and compute the step p_k^{LM}.

2. Check the noise level. If noise is too high reduce it.
3. Step acceptance based on $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k(x_k) - m_k(x_k + p_k^{LM})}$.
4. Regularization parameter update.

We propose a **Levenberg-Marquardt method**.

Algorithm : k-th iteration

1. Step computation: define the LM model

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k)+J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

   and compute the step p_k^{LM}.
2. Check the noise level. If noise is too high reduce it.
3. Step acceptance based on $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k)-f_{\delta_k}(x_k+p_k^{LM})}{m_k(x_k)-m_k(x_k+p_k^{LM})}$.
4. Regularization parameter update.

1) The step

- The step is the solution of the linearized least squares subproblem:

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

where λ_k is an appropriately chosen regularization parameter.

- This is equivalent to:

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k = -g_{\delta_k}(x_k)$$

1) The step

- The step is the solution of the linearized least squares subproblem:

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

where λ_k is an appropriately chosen regularization parameter.
- This is equivalent to:

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k = -g_{\delta_k}(x_k) + r_k.$$

- Large scale problems: an inexact step is computed.

1) The step

- The step is the solution of the linearized least squares subproblem:

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

where λ_k is an appropriately chosen regularization parameter.

- This is equivalent to:

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k = -g_{\delta_k}(x_k) + r_k.$$

- Large scale problems: an inexact step is computed.
- For a residual, $\|r_k\| \leq \epsilon_k \|g_{\delta_k}\|$ with ϵ_k small enough, the step achieves the Cauchy decrease:

$$m_k(x_k) - m_k(x_k+p) \geq \frac{\theta}{2}\frac{\|g_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}, \qquad \theta > 0.$$

which is sufficient to get global convergence.

We propose a **Levenberg-Marquardt method**.

Algorithm : k-th iteration

1. Step computation: define the LM model

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

   and compute the step p_k^{LM}.
2. Check the noise level. If noise is too high reduce it.
3. Step acceptance based on $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k)-f_{\delta_k}(x_k+p_k^{LM})}{m_k(x_k)-m_k(x_k+p_k^{LM})}$.
4. Regularization parameter update.

2) Noise control

- The optimization process starts with a given noise level $\delta = \delta_0$ depending on $|X_0|$.
- **Noise control**: our method relies on a mechanism to control the noise: if it is judged to be too large it is reduced.
- We assume to have access to function and gradient values at every accuracy level.
- The noise is driven to zero along the optimization process.

Assumption

It exists $\bar{K} > 0$ and $\delta_k \geq 0$, such that:

$$|f_{\delta_k}(x) - f(x)| = \left|\frac{1}{2}\|F_{\delta_k}(x)\|^2 - \frac{1}{2}\|F(x)\|^2\right| \leq \delta_k,$$

$$\|g(x) - g_{\delta_k}(x)\| \leq \bar{K}\delta_k.$$

2) Noise control

- Given the noise level δ_k, in [Trust region methods, Conn, Gould, Toint] this condition is used:

$$\delta_k \leq \eta_0 [m_k(x_k) - m_k(x_k + p_k^{LM})],$$

  with η_0 appropriately chosen, to ensure a true reduction in the noise-free objective function f.

- $m_k(x_k) - m_k(x_k + p_k^{LM}) = O(\lambda_k \| p_k^{LM} \|^2)$.
- Noise control:

$$\delta_k \leq \kappa_d \lambda_k^{\alpha} \| p_k^{LM} \|^2,$$

  for suitable constants $\kappa_d > 0$ and $\alpha \in \left[\frac{1}{2}, 1\right)$.

- The noise tends to zero:

$$\lim_{k \to \infty} \lambda_k \| p_k^{LM} \|^2 = 0.$$

We propose a **Levenberg-Marquardt method**.

Algorithm : k-th iteration

1. Step computation: define the LM model

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

   and compute the step p_k^{LM}.
2. Check the noise level. If noise is too high reduce it.
3. Step acceptance based on $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k+p_k^{LM})}{m_k(x_k) - m_k(x_k+p_k^{LM})}$.
4. Regularization parameter update.

3) Step acceptance

Step acceptance based on ratio between actual and predicted reduction:

$$\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k(x_k) - m_k(x_k + p_k^{LM})}.$$

1. If $\rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1$, accept the step $x_{k+1} = x_k + p_k^{LM}$,
2. Otherwise reject the step $x_{k+1} = x_k$.

We propose a **Levenberg-Marquardt method**.

Algorithm : k-th iteration

1. Step computation: define the LM model

$$\min_{p\in\mathbb{R}^n} m_k(x_k+p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k\|p\|^2,$$

   and compute the step p_k^{LM}.
2. Check the noise level. If noise is too high reduce it.
3. Step acceptance based on $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k+p_k^{LM})}{m_k(x_k) - m_k(x_k+p_k^{LM})}$.
4. Regularization parameter update.

4) Parameter update

The parameter update is inspired by [Bergou, Gratton, Vicente, 2016] and [Bandeira, Scheinberg, Vicente, 2014].
Given $\gamma > 1$

- Successful step:

$$\lambda_{k+1} = \begin{cases} \min\{\gamma\lambda_k, \lambda_{\max}\} & \text{if } \|g_{\delta_k}(x_k)\| < \eta_2/\lambda_k, \\ \lambda_k & \text{if } \|g_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k. \end{cases}$$

- Unsuccessful step:

$$\lambda_{k+1} = \gamma\lambda_k.$$

We increase the parameter even in case of successful iterations.

4) Parameter update

The parameter update is inspired by [Bergou, Gratton, Vicente, 2016] and [Bandeira, Scheinberg, Vicente, 2014].
Given $\gamma > 1$

- Successful step:

$$\lambda_{k+1} = \begin{cases} \min\{\gamma\lambda_k, \lambda_{\max}\} & \text{if } \|g_{\delta_k}(x_k)\| < \eta_2/\lambda_k, \\ \lambda_k & \text{if } \|g_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k. \end{cases}$$

- Unsuccessful step:

$$\lambda_{k+1} = \gamma\lambda_k.$$

We increase the parameter even in case of successful iterations.

$$\frac{\|g(x_k)\|}{(1+c_k)} \leq \|g_{\delta_k}(x_k)\| \leq \frac{\|g(x_k)\|}{(1-c_k)}, \text{ with } c_k = O\left(\frac{1}{\lambda_k^{1-\alpha/2}}\right).$$

Theoretical results

Assumptions

- Assumption 1:
  Function f is continuously differentiable, and it exists $\kappa_J > 0$ such that for all $k \geq 0$ and all $x \in [x_k, x_k + p_k^{LM}]$, $\|J_\delta(x)\| \leq \kappa_J$.
- Assumption 2: f has Lipschitz continuous gradient:
  $\|g(x) - g(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.

Global Convergence

Let the residual be small enough, i.e. r_k satisfies $\|r_k\| \leq \epsilon_k \|g_{\delta_k}\|$, with

$$\epsilon_k \leq \min \left\{ \frac{\theta_1}{\lambda_k^\alpha}, \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}} \right\}$$

where $\theta_1 > 0$, $\theta_2 \in \left(0, \frac{1}{2}\right]$ and $\alpha \in \left[\frac{1}{2}, 1\right)$.

Lemma

The sequences $\{\delta_k\}$ and $\{x_k\}$ generated by the Algorithm are such that

$$\lim_{k\to\infty} \delta_k = 0, \qquad \lim_{k\to\infty} \|g(x_k)\| = 0.$$

Local convergence

Asymptotic step behaviour

The LM step asymptotically tends to the direction of the negative perturbed gradient:

$$\lim_{k\to\infty}(p_k^{LM})_i + \frac{\theta}{\kappa_J^2+\lambda_k}(g_{\delta_k}(x_k))_i = 0 \quad \text{for} \quad i = 1,\dots,n,$$

where $(\cdot)_i$ denotes the i-th vector component.

Lemma

Let $p_k^{SD} = -\frac{\theta}{\kappa_J^2+\lambda_k} g_{\delta_k}(x_k)$. *If* $x_{\bar{k}} \in B_r(x^*)$ *and* $\lambda_{\bar{k}}$ *big enough,*

- $\|x_{k+1} - x^*\| < \|x_k - x^*\|$, *for all* $k \geq \bar{k}$.
- $\|x_k - x^*\|$ *tends to zero.*

Complexity analysis

Assumption

Let assume that the procedure is stopped when $\|g_{\delta_k}(x_k)\| \leq \epsilon$.

- The number of successful iterations N_1 is bounded above by:

$$N_1 \leq f_{\delta_{k_s-1}}(x_{k_s}) \frac{2}{\eta_1} \frac{\kappa_J^2 + \lambda_{\max}}{\theta \epsilon^2} = O(\epsilon^{-2}).$$

- The number of unsuccessful iterations N_3 is bounded above by a constant independent of ϵ:

$$N_3 \leq \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma}.$$

Complexity

Standard Levenberg-Marquardt methods complexity is preserved:

$$N_T = O(\epsilon^{-2}),$$

Numerical Results

Test problems

We consider two problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 + \frac{1}{2}\|x\|^2 = \sum_{j=1}^{N} F_j(x)^2 + \frac{1}{2}\|x\|^2,$$

with $F_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, \ldots, N$, N total number of samples.

Numerical Results

Test problems

We consider two problems of the form

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 + \frac{1}{2}\|x\|^2 = \sum_{j=1}^{N} F_j(x)^2 + \frac{1}{2}\|x\|^2,$$

with $F_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, \dots, N$, N total number of samples.

- P1: Data assimilation problem
- P2: Machine learning problem

Approximations

- Function approximations built through a random subsampling.
- $J_{\delta_k}(x) \in \mathbb{R}^{K_k \times n}$ is the Jacobian matrix of $F_{\delta_k}(x)$.
- $g_{\delta_k} \in \mathbb{R}^n$ the gradient of f_{δ_k}.

Linear algebra phase

- CG method.
- $\|r_k\| \leq 10^{-1}\|g_{\delta_k}(x_k)\|$

Performance evaluation criteria

We compare subsampled Levenberg-Marquardt method **(SSLM)** and full Levenberg-Marquardt method **(FLM)** ($K_k = N, \forall k$).

Cost counters

We evaluate savings arising from the employment of the noise control strategy.

- $cost_f$ weighted counter of function evaluations costs (if $|X_k| = N$ cost=1, if $|X_k| = K_k$ cost=K_k/N.) $\rightarrow$ $save_f$ savings in function evaluations.
- $cost_p$ weighted counter of products costs (if $|X_k| = N$ cost=1, if $|X_k| = K_k$ cost=K_k/N.) $\rightarrow$ $save_p$ savings in products.

Given the current sample set X_k, s.t. $|X_k| = K_k$.

Noise update

Given the step, check the noise: $\delta_k \leq \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2$?
If not, repeat:

1. Increase the samples set size: $|X_{k+1}| = K_* |X_k|$.
2. Recompute function, Jacobian and gradient.
3. Need to check condition again $\rightarrow$ Need to recompute the step: $(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k = -g_{\delta_k}(x_k) + r_k$.

$\rightarrow$ Resulting samples set size: $|X_{k+1}| = K_*^{n_k} |X_k|$.

Given the current sample set X_k, s.t. $|X_k| = K_k$.

Noise update

Given the step, check the noise: $\delta_k \leq \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2$?
If not, repeat:

1. Increase the samples set size: $|X_{k+1}| = K_* |X_k|$.
2. Recompute function, Jacobian and gradient.
3. Need to check condition again $\rightarrow$ Need to recompute the step: $(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k = -g_{\delta_k}(x_k) + r_k$.

$\rightarrow$ Resulting samples set size: $|X_{k+1}| = K_*^{n_k} |X_k|$.

Parameters affecting the cost

- $\delta_k \leq \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2$.
- K_0 cardinality of the starting sample set.
- $|X_{k+1}| = K_*^{n_k} |X_k|$.

P1: Data assimilation problem

Nonlinear wave equation:

$$\frac{\partial^2 u(z,t)}{\partial t^2} - \frac{\partial^2 u(z,t)}{\partial z^2} + \mu e^{\nu u} = 0,$$
$$u(0,t) = u(1,t) = 0,$$
$$u(z,0) = u_0(z),\ \frac{\partial u(z,0)}{\partial t} = 0,$$
$$0 \le t \le T,\ 0 \le 0 \le 1.$$

- We look for the initial state $u_0(z)$, from the knowledge of observations $u(z_i, t_j),\ t_j > 0$.
- We consider a mesh involving $n = 360$ grid points for the spatial discretization and $N_t = 64$ for the temporal one.
- We assume to have an observation at each grid point: $N = n \times N_t = 23040$.

P1: Data assimilation problem

It is possible to recover $u_0(z)$ solving the following data assimilation problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N_t} \|H_j(x(t_j)) - y_j\|^2_{R_j^{-1}}$$

- $\|x\|^2_M = x^T M x$ for a symmetric positive definite matrix M,
- $x_b \in \mathbb{R}^n$ is the background vector (a priori estimate)
- $y_j \in \mathbb{R}^{m_j}$ is the vector of observations at time t_j, $m_j \leq n$.
- H_j is the operator modelling the observation process at t_j
- $x(t_j)$ the state vector, solution of the nonlinear model at time t_j.

P1: Data assimilation problem

- Background vector and observations from a chosen initial true state by adding noise $N(0, \sigma_b^2)$ and $N(0, \sigma_o^2)$ with $\sigma_b = 0.2$, $\sigma_o = 0.05$.
- Covariances matrices are diagonal: $B = \sigma_b^2 I_n$ and $R_j = \sigma_o^2 I_{m_j}$ $\forall j$.
- Least-squares problem reformulation:

$$F(x) = \begin{bmatrix} \frac{1}{\sigma_o}(H_0(x(t_0)) - y_0) \\ \vdots \\ \frac{1}{\sigma_o}(H_{N_t}(x(t_{N_t})) - y_{N_t}) \end{bmatrix}$$

where $(H_j(x(t_j)) - y_j) \in \mathbb{R}^{m_j}$ for $j = 1, \dots, N_t$.

P1: Data assimilation problem

- Background vector and observations from a chosen initial true state by adding noise $N(0, \sigma_b^2)$ and $N(0, \sigma_o^2)$ with $\sigma_b = 0.2$, $\sigma_o = 0.05$.
- Covariances matrices are diagonal: $B = \sigma_b^2 I_n$ and $R_j = \sigma_o^2 I_{m_j}\ \forall j$.
- Least-squares problem reformulation:

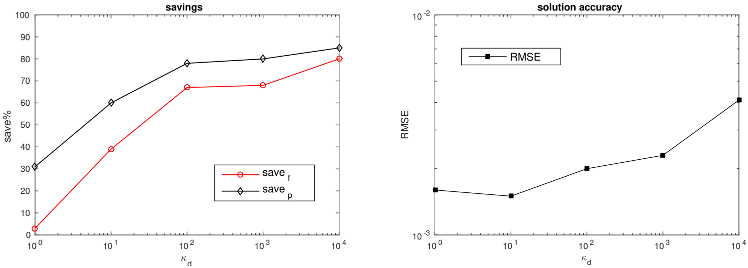
$$F(x) = \begin{bmatrix} \frac{1}{\sigma_o}(H_0(x(t_0)) - y_0) \\ \vdots \\ \frac{1}{\sigma_o}(H_{N_t}(x(t_{N_t})) - y_{N_t}) \end{bmatrix}$$

where $(H_j(x(t_j)) - y_j) \in \mathbb{R}^{m_j}$ for $j = 1, \ldots, N_t$.

- Kept $K_* = 1.5$ fixed, we study the effect of κ_d, depending on K_0.

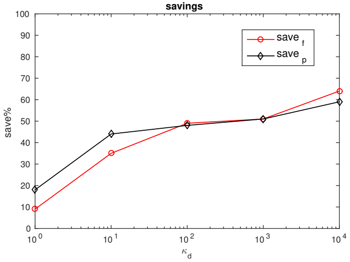
P1: effect of κ_d

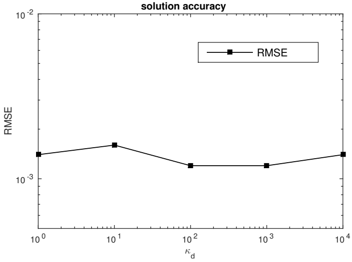
$K_0 = 2000$, $K_* = 1.5$	FLM	SSLM $\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	12	12	12	11
$\mathbf{CG_{it}}$	2.4	5.4	4.9	4.2	4.2	3.9
$\mathbf{cost_f}$	10	9.7	6.1	3.3	3.2	2.0
$\mathbf{cost_p}$	67	46.1	26.8	14.9	13.5	10.3
$\mathbf{\lvert X_{it} \rvert}$	23040	15188	6750	3000	3000	2000
RMSE	1.2e-2	3.0e-2	2.8e-2	3.8e-2	4.4e-2	7.8e-2
$\mathbf{save_f}$		3%	39%	67%	68%	80%
$\mathbf{save_p}$		31%	60%	78%	80%	85%



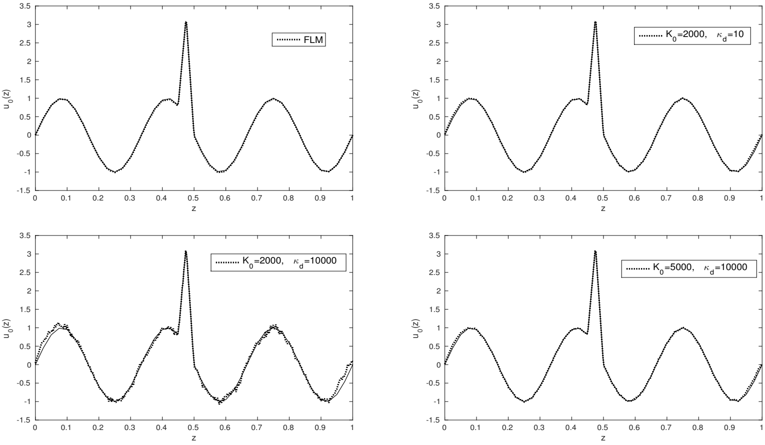
P1: savings vs solution accuracy

$K_0 = 5000$ $K_* = 1.5$	FLM	SSLM				
		$\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	11	12	12	12
$\mathbf{CG_{it}}$	2.4	4.1	3.9	4.0	4.1	3.7
$\mathbf{cost_f}$	10	9.1	6.5	5.1	4.9	3.6
$\mathbf{cost_p}$	67	54.8	37.2	34.6	32.9	27.3
$\mathbf{\lvert X_{it} \rvert}$	23040	16875	11250	7500	7500	5000
RMSE	1.2e-2	2.7e-2	3.0e-2	2.1e-2	2.1e-2	2.7e-2
$\mathbf{save_f}$		9%	35%	49%	51%	64%
$\mathbf{save_p}$		18%	44%	48%	51%	59%





P1: solution approximations



P2: Machine learning problem

Binary classification problem: $\{(z^i, y^i)\}$ with $z^i \in \mathbb{R}^n$, $y^i \in \{-1, +1\}$ and $i = 1, \dots, N$.
Training objective function: logistic loss with l_2 regularization

$$f(x) = \frac{1}{2N} \sum_{i=1}^{N} \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2N} \|x\|^2.$$

Least-squares form:

$$F(x) = \frac{1}{N} \begin{bmatrix} \sqrt{\log(1 + \exp(-y^1 x^T z^1))} \\ \vdots \\ \sqrt{\log(1 + \exp(-y^N x^T z^N))} \end{bmatrix}.$$

P2: machine learning problem

Approximations to f are built as:

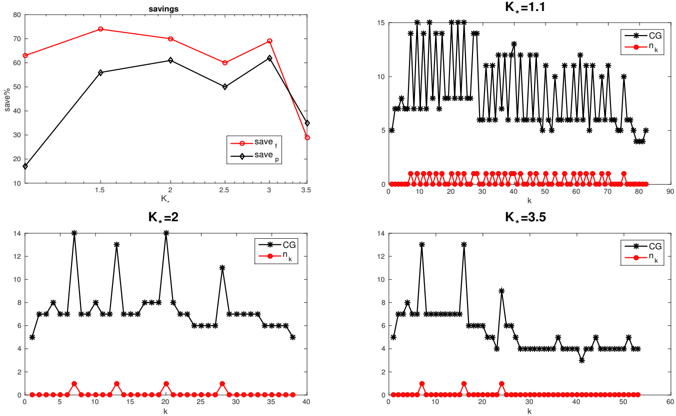
$$f_{\delta_k}(x) = \frac{1}{2K_k} \sum_{i \in X_k} \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2K_k} \|x\|^2.$$

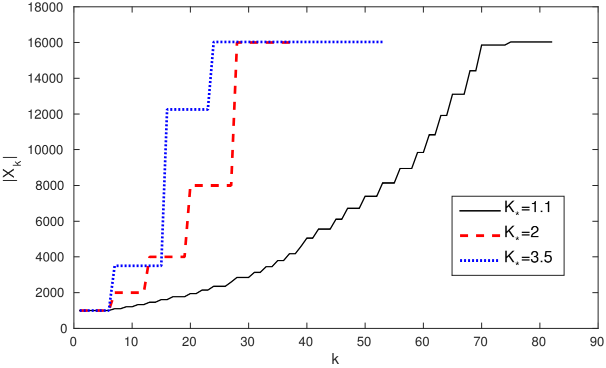
We consider the **CINA dataset** [http://www.causality.inf.ethz.ch/data/CINA.html], for which $n = 132$, $N = 16033$ for the training set, $\tilde{N} = 10000$ for the testing set.

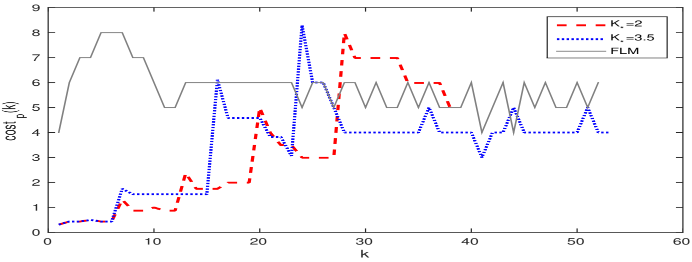
Noise control condition parameters

- $K_0 = 132$.
- $\kappa_d = 10$.
- We study the effect of K_*.

	FLM	SSLM					
		$K_* = 1.1$	$K_* = 1.5$	$K_* = 2$	$K_* = 2.5$	$K_* = 3$	$K_* = 3.5$
it	52	82	43	38	39	34	53
CG_{it}	5.7	8.5	8.0	7.5	7.3	7.2	5.5
$cost_f$	53	19.8	14.1	15.9	21.2	16.5	37.7
$cost_p$	808	671.2	351.3	316.7	400.7	310.4	521.1
RMSE	6.0e-2	1.0e-1	6.6e-2	5.4e-2	4.7e-2	4.1e-2	3.9e-2
$save_f$		63%	74%	70%	60%	69%	29%
$save_p$		17%	56%	61%	50%	62%	35%




18000
16000
14000
12000
10000
8000
6000
4000
2000
0
|X_k|
0
10
20
30
40
50
60
70
80
90
k
K*=1.1
K*=2
K*=3.5


9
8
7
6
5
4
3
2
1
0
cost_p(k)
0
10
20
30
40
50
60
k
K*=2
K*=3.5
FLM

THANK YOU FOR YOUR ATTENTION!

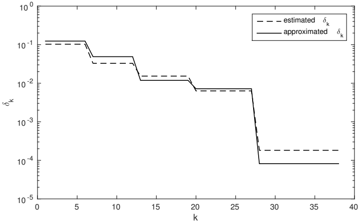
Additional Assumption

- Let f be twice differentiable in an open set containing $\mathcal{L}$,
- $H(x^*) \succeq 0$, H Hessian matrix of f,
- $\|H(x) - H(y)\| \leq M\|x - y\|$ for all $x, y \in \mathcal{L}$,
- $0 < l \leq L < \infty$ such that $l\ I_n \preceq H(x^*) \preceq L\ I_n$ with I_n the identity matrix of size n.

We estimate the noise in the following way:

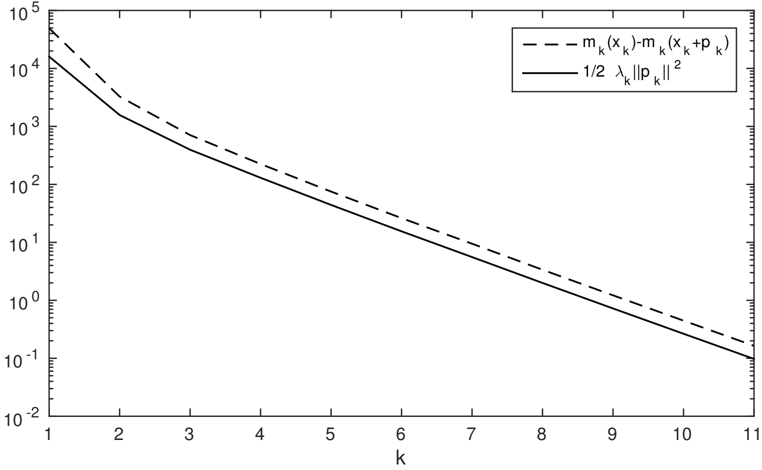
$$\delta_k \simeq \frac{\sqrt{2(N-K_k)}}{K_k}, \quad \text{with} \quad K_k = |X_k|.$$

If the components $F_i(x)$ of $F(x)$ were Gaussian, $\sum_{i=1}^{N-K_k} F_i(x)^2$ would follow a Chi-squared distribution with standard deviation $\sqrt{2(N-K_k)}$.



Solver	it	CG_{it}	$cost_f$	$cost_p$	$\lvert X_{it} \rvert$	err	e_{te}
$SSLM_{est}$	38	7.5	15.9	316.7	16000	5.4e-2	0.187
$SSLM_{appr}$	37	7.4	17.7	318.1	16000	5.7e-2	0.186

Model



$m_k(x_k)$-$m_k(x_k+p_k)$
1/2 $\lambda_k||p_k||^2$
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
1
2
3
4
5
6
7
8
9
10
11
k

Noisy vs exact gradient

For λ_k sufficiently large it exists $c_k \in (0,1)$ such that

$$\frac{\|g(x_k)\|}{(1+c_k)} \leq \|g_{\delta_k}(x_k)\| \leq \frac{\|g(x_k)\|}{(1-c_k)}, \text{ with } c_k = \frac{2\bar{K}\sqrt{\kappa_d}}{\lambda_k^{1-\alpha/2}}.$$

Gradient approximation

For λ_k large $\rightarrow$ $\|g_{\delta_k}(x_k)\| \simeq \|g(x_k)\|$.

The quality of the approximations of f and g at x depends on the distance $\max\{\|F_\delta(x) - F(x)\|, \|J_\delta(x) - J(x)\|\}$, as follows:

$$\begin{aligned} |f_{\delta_k}(x) - f(x)| &\leq \frac{1}{2}\|F_\delta(x) - F(x)\| \sum_{j=1}^{N} |F_j(x) + (F_\delta)_j(x))|, \\ \|g(x) - g_{\delta_k}(x)\| &\leq \|J_\delta(x) - J(x)\|\|F(x)\| + \|J_\delta(x)\|\|F_\delta(x) - F(x)\|. \end{aligned}$$

Then, we can assume that there exist $\bar{K} \geq 0$ and $\delta_k \geq 0$, such that at each iteration k uniformly in x:

$$|f_{\delta_k}(x) - f(x)| = \left| \frac{1}{2}\|F_{\delta_k}(x)\|^2 - \frac{1}{2}\|F(x)\|^2 \right| \leq \delta_k, \tag{1}$$

$$\|g(x) - g_{\delta_k}(x)\| \leq \bar{K}\delta_k. \tag{2}$$

We will refer to δ_k as to the noise level.