Efficient Solution of Sparse Optimization Problems via Interior Point Methods

Daniela di Serafino

Dip. Matematica e Applicazioni, Univ. Napoli Federico II, Italy

daniela.diserafino@unina.it

Università degli Studi di Napoli Federico II

Dipartimento di Matematica e Applicazioni "Renato Caccioppoli"

Modern Techniques of Very Large Scale Optimization

The University of Edinburgh, May 18-19, 2022

Acknowledgments

Co-authors

Valentina De Simone
Univ. Campania Vanvitelli, IT

Jacek Gondzio
Univ. Edinburgh, UK

Spyros Pougkakiotis
Univ. Edinburgh, UK
(now at Yale Univ., USA)

Marco Viola
Univ. Campania Vanvitelli, IT

Funding

- Istituto Nazionale di Alta Matematica - Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS), Italy
- V:ALERE Program of the University of Campania “Luigi Vanvitelli”, Italy
- University of Edinburgh
- Google Project *Fast* $(1+x)$*-order Methods for Linear Programming*

Problem and goal

Efficient solution of a class of optimization problems which are very large and are expected to yield sparse solutions

$$\begin{array}{ll} \min\limits_{x} & f(x) + \tau_1 \|x\|_1 + \tau_2 \|Lx\|_1 \\ \text{s.t.} & Ax = b \end{array}$$

$f : \mathbb{R}^n \to \mathbb{R}$ twice continuously differentiable convex function, $L \in \mathbb{R}^{l \times n}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m \leq n$, and $\tau_1, \tau_2 > 0$

$\|x\|_1$ and $\|Lx\|_1$ induce sparsity in x and/or in some dictionary Lx

- Many applications: portfolio optimization, signal/image processing, classification in statistics and machine learning, inverse problems, compressed sensing, ...
- Usually solved by specialized first-order methods, but those methods may be too expensive or struggle with not-so-well conditioned problems

Problem and goal (cont'd)

Non-smooth second-order methods:

- proximal (projected) Newton-type methods
- semi-smooth Newton methods combined with augmented Lagrangian methods

Our goal:

show that Interior Point Methods (IPMs) can be equally or more efficient, robust and reliable than well-assessed first-order methods, by

- exploiting problem features in the linear algebra phase of IPMs
- taking advantage of the expected sparsity of the optimal solution

Applications used to support our view

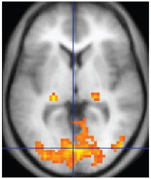
Multi-period portfolio optimization: computing the optimal investment on a basket of s assets, over medium- and long-time horizons, allowing rebalancing at intermediate periods based on available information

(Generalized) Fused LASSO

$$\min_w \ \frac{1}{2} w^T C w + \tau_1 \|w\|_1 + \tau_2 \|Lw\|_1 \quad \text{s.t.} \ \ A\,w = b$$

$$\left(w^T = [w_1^T, \ldots, w_m^T], \quad Lw = \textstyle\sum_{j=1}^{m-1} \|w_{j+1} - w_j\|_1\right)$$

Binary classification of functional Magnetic Resonance Imaging (fMRI) data:



(Wikipedia)

using BOLD measures of brain spatio-temporal activity, train a linear classifier to distinguish between different classes of patients (e.g. ill/healthy) or different kinds of stimuli (e.g. pleasant/unpleasant) and get information on the most significant brain areas

ℓ1-TV-regularized Least Squares (3D Fused LASSO)

$$\min_w \ \frac{1}{2s} \|Dw - y\|^2 + \tau_1 \|w\|_1 + \tau_2 \|Lw\|_1$$

($\|Lw\|_1$ discrete anisotropic TV)

Applications used to support our view (cont'd)

TV-based Poisson Image Restoration: denoising and deblurring of images corrupted by Poisson noise (fluorescence microscopy, computed tomography, astronomical imaging, ...)

blurry and noisy

Restored image - IP-PMM

Regularized Kullback-Leibler Divergence

$$\begin{array}{ll} \min\limits_{w} & KL(Dw + a, g) + \lambda \|Lw\|_1 \\ \text{s.t.} & e_n^\top w = r, \; w \geq 0 \end{array}$$

(L discrete isotropic TV)

Linear Classification via Logistic Regression: training a linear binary classifier by using the logistic model

Regularized Logistic Loss

$$\min_{w} \; \phi(w) + \tau \|w\|_1, \quad \phi(w) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(w) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-g^i w^\top d^i}\right)$$

Remaining part of this talk

- Interior Point Methods (IPMs) for convex programming
- Interior Point-Proximal Method of Multipliers (IP-PMM)
- Applications:
  - Portfolio Selection
  - Binary Classification of fMRI data
  - TV-based Poisson Image Restoration
  - Linear Classification via Regularized Logistic Regression

  For each application: efficient linear algebra solvers, variable dropping techniques to take advantage of sparsity in the solution, numerical results and comparisons with first-order methods
- Conclusions

Modeling trick

Original formulation

$$\begin{array}{ll} \min\limits_{x} & f(x) + \tau_1\|x\|_1 + \tau_2\|Lx\|_1 \\ \text{s.t.} & Ax = b \end{array} \qquad L \in \mathbb{R}^{l\times n},\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m,\ m \le n$$

For any a, let $|a| = a^+ + a^-$, where $a^+ = \max\{a, 0\}$ and $a^- = \max\{-a, 0\}$
Set $d = Lx \in \mathbb{R}^l$

New formulation

$$\begin{array}{ll} \min\limits_{x^+,x^-,d^+,d^-} & f(x^+ - x^-) + \tau_1(e_n^\top x^+ + e_n^\top x^-) + \tau_2(e_l^\top d^+ + e_l^\top d^-) \\ \text{s.t.} & A(x^+ - x^-) = b \\ & L(x^+ - x^-) = d^+ - d^- \\ & x^+, x^-, d^+, d^- \ge 0 \end{array}$$

$e_j \in \mathbb{R}^j$ vector of all 1's

Larger smooth problem, but IPMs are able to efficiently handle large sets of linear equality and non-negativity constraints!

(Primal-dual) IPMs for convex programming

Problem in standard form: $\min_x f(x), \quad \text{s.t.} \;\; Ax = b, \;\; x \geq 0$

Basic ideas of IPMs

- handle non-negativity constraints with a logarithmic barrier in the objective function
- approximately solve a sequence of barrier problems by using a (possibly inexact) Newton method

(Primal-dual) IPMs for convex programming

Problem in standard form: $\min_x f(x), \quad \text{s.t. } Ax = b, \quad x \geq 0$

Basic ideas of IPMs

- handle non-negativity constraints with a logarithmic barrier in the objective function
- approximately solve a sequence of barrier problems by using a (possibly inexact) Newton method

At each iteration k

- barrier problem: $\min_x f(x) - \mu_k \sum_{j=1}^{n} \ln x^j, \quad \text{s.t. } Ax = b \quad (\mu_k > 0)$
- Newton system: $\begin{bmatrix} -(\nabla^2 f(x_k) + \Theta_k^{-1}) & A^\top \\ A & 0_{m,m} \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} \bar{r}_{1,k} \\ \bar{r}_{2,k} \end{bmatrix}$

  $\Theta_k = X_k Z_k^{-1}$, $X_k = \mathrm{diag}(x_k)$, $Z_k = \mathrm{diag}(z_k)$, $x_k, z_k > 0$

(Primal-dual) IPMs for convex programming (cont'd)

- As $\mu_k \to 0$, an optimal solution of the barrier problem converges to an optimal solution of the original problem [Wright S., book 1997; Forsgren, Gill & Wright M., SIREV 2002]

- Polynomial convergence with respect to the number of variables has been proved for various classes of problems [Nesterov & Nemirovskii, SIAM Studies Appl Math 1994; Zhang, SIOPT 1994]

- Θ_k contains some very large and some very small elements close to optimality
  $\Longrightarrow$ the KKT matrix becomes increasingly ill-conditioned
  $\Longrightarrow$ regularization is beneficial
  [Friedlander, SIOPT 2007; D'Apuzzo, De Simone & **dS**, COAP 2010; Gondzio, EJOR 2012]

- The augmented system can be solved either directly (by an appropriate factorization) or iteratively (by an appropriate Krylov subspace method) [D'Apuzzo, De Simone & **dS**, COAP 2010; Gondzio, EJOR 2012; **dS** & Orban, SISC 2021]

Regularization in IPMs

Use regularization to improve the spectral properties of the KKT matrix

- Dual regularization $\rightarrow$ (2,2) block:

$$0_{m,m} + \delta_k I_m, \quad \delta_k > 0 \quad ([A\ \delta I_m] \text{ full rank})$$

- Primal regularization $\rightarrow$ (1,1) block:

$$\nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I_n, \quad \rho_k > 0 \quad (\text{eigs bounded away from } 0)$$

A natural way of introducing regularization is through the use of proximal point methods [Altman & Gondzio, OMS 1999; Friedlander & Orban, Math Program Comput 2012; Pougkakiotis & Gondzio, COAP 2021]

This (algorithmic) regularization allows us to retrieve the solution of the original problem

Interior Point - Proximal Method of Multipliers (IP-PMM)

Merge IPM with PMM [Pougkakiotis & Gondzio, COAP 2021]

Problem formulation (equivalent to the standard one):

$$\min_x \ f(x), \quad \text{s.t.} \ \ Ax = b, \ \ x^{\mathcal{I}} \geq 0, \ \ x^{\mathcal{F}} \ \text{free}$$

$\mathcal{I} \subseteq \{1, \ldots, n\}$, $\mathcal{F} = \{1, \ldots, n\} \setminus \mathcal{I}$

Iteration k: given an estimate η_k for an optimal Lagrange multiplier vector y^* associated to $Ax = b$ and an estimate ζ_k of a primal solution x^*

- PMM: minimize the proximal penalty function $(\rho_k, \delta_k > 0)$

$$\mathcal{L}^{PMM}_{\rho_k,\delta_k}(x; \zeta_k, \eta_k) = f(x) - \eta_k^\top (Ax - b) + \frac{1}{2\delta_k} \|Ax - b\|_2^2 + \frac{\rho_k}{2} \|x - \zeta_k\|_2^2$$

- IP-PMM: solve the PMM subproblem by applying one or more iters of IPM, i.e. alter the proximal penalty function with a barrier

$$\mathcal{L}^{IP-PMM}_{\rho_k,\delta_k}(x; \zeta_k, \eta_k) = \mathcal{L}^{PMM}_{\rho_k,\delta_k}(x; \zeta_k, \eta_k) - \mu_k \sum_{j \in \mathcal{I}} \ln x^j$$

IP-PMM: Newton system

By writing the optimality conditions, applying a Newton step and performing straightforward computations we get the (symmetric indefinite) regularized augmented system

$$\begin{bmatrix} -(\nabla^2 f(x_k) + \Xi_k + \rho_k I_n) & A^\top \\ A & \delta_k I_m \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{1,k} \\ r_{2,k} \end{bmatrix}$$

$$\Xi_k = \begin{bmatrix} 0_{|\mathcal{F}|,|\mathcal{F}|} & 0_{|\mathcal{I}|,|\mathcal{F}|} \\ 0_{|\mathcal{F}|,|\mathcal{I}|} & (X_k^{\mathcal{I}})^{-1}(Z_k^{\mathcal{I}}) \end{bmatrix}$$

In some cases (e.g. $\nabla^2 f(x_k)$ zero or diagonal) it is convenient to eliminate Δx, obtaining the (symmetric positive definite - spd) regularized normal equations

$$\left(A(\nabla^2 f(x_k) + \Xi_k + \rho_k I_n)^{-1} A^\top + \delta_k I_m\right) \Delta y = r$$

Application 1: multi-period portfolio optimization

- Investment period partitioned into m sub-periods $[t^j, t^{j+1})$, decisions taken at each t_j
- Portfolio defined by $w = [w_1^\top, w_2^\top, \ldots, w_m^\top]^\top$ ($w_j \in \mathbb{R}^s$ portfolio at t^j, s # assets)
- Markowitz-type model: minimize the sum of the risks over the periods
- Asset correlation (ill-conditioned covariance matrices of returns), few active positions i.e. vars > 0 (reduction of holding costs), small changes of active positions (reduction of transaction costs) $\Longrightarrow$ regularization, sparse and "smooth" solutions

Application 1: multi-period portfolio optimization

- Investment period partitioned into m sub-periods $[t^j, t^{j+1})$, decisions taken at each t_j
- Portfolio defined by $w = [w_1^\top, w_2^\top, \ldots, w_m^\top]^\top$ $(w_j \in \mathbb{R}^s$ portfolio at t^j, s # assets)
- Markowitz-type model: minimize the sum of the risks over the periods
- Asset correlation (ill-conditioned covariance matrices of returns), few active positions i.e. vars > 0 (reduction of holding costs), small changes of active positions (reduction of transaction costs) $\Longrightarrow$ regularization, sparse and "smooth" solutions

$$\begin{array}{lll} \min\limits_{w} & \frac{1}{2} w^\top C\, w + \tau_1 \|w\|_1 + \tau_2 \|Lw\|_1, \qquad \tau_1, \tau_2 > 0 & Lw = \sum_{j=1}^{m-1} \|w_{j+1} - w_j\|_1 \\ \text{s.t.} & \left.\begin{array}{l} w_1^\top e_s = \xi_{\text{init}} \\ w_j^\top e_s = (e_s + r_{j-1})^\top w_{j-1}, \quad j = 2, \ldots, m \\ (e_s + r_m)^\top w_m = \xi_{\text{term}} \end{array}\right\} & \overline{A} w = \overline{b} \end{array}$$

$n = m\, s$, $C = \text{diag}(C_1, C_2, \ldots, C_m) \in \mathbb{R}^{n \times n}$ block-diag spd, $L \in \mathbb{R}^{(n-s) \times n}$ fused-lasso operator, $r_j \in \mathbb{R}^s$ expected return at t^j, ξ_{init} initial wealth, ξ_{term} target wealth

[Corsaro, De Simone & Marino, Ann Oper Res 2019]

Application 1: multi-period portfolio optimization (cont'd)

Smooth problem reformulation

$$\min_x \ \frac{1}{2} x^\top Q x + c^\top x \quad \text{s.t.} \quad Ax = b, \quad x \geq 0$$

$$d = Lw, \quad x = [(w^+)^\top, (w^-)^\top, (d^+)^\top, (d^-)^\top]^\top$$

$$Q = \begin{bmatrix} \begin{bmatrix} C & -C \\ -C & C \end{bmatrix} & 0_{2n,2l} \\ 0_{2l,2n} & 0_{2l,2l} \end{bmatrix}, \quad A = \begin{bmatrix} \overline{A} & -\overline{A} & 0_{(m+1),2l} \\ L & -L & \begin{bmatrix} -I_l & I_l \end{bmatrix} \end{bmatrix}$$

$$c = [\tau_1, \ldots, \tau_1, \tau_2, \ldots, \tau_2]^\top \in \mathbb{R}^{\overline{n}}, \quad b = [\overline{b}^1, \ldots, \overline{b}^{m+1}, 0, \ldots, 0]^\top \in \mathbb{R}^{\overline{m}}$$

$$l = n - s, \ \overline{n} = 2(n + l) = 2s(2m - 1), \ \overline{m} = m + 1 + l = (m + 1) + s(m - 1)$$

Portfolio optimization: dropping & linear system solution

The optimal solution is expected to be (and actually is) sparse $\Longrightarrow$ dropping strategy:

- set a threshold $\epsilon_{\mathrm{drop}} > 0$ and a large constant $\xi > 0$
- iter $k = 0$: set $\mathcal{V} = \emptyset$
- iter $k > 0$: for every $j \in \mathcal{I} \setminus \mathcal{V}$, drop (i.e. set to 0) x_k^j and z_k^j such that

$$x_k^j \leq \epsilon_{\mathrm{drop}} \quad \text{and} \quad z_k^j \geq \xi \cdot \epsilon_{\mathrm{drop}} \quad \text{and} \quad (r_d)_k^j \leq \epsilon_{\mathrm{drop}}$$

and set $\mathcal{V} = \mathcal{V} \cup \{j\}$ (dropped indices), $\mathcal{G} = \mathcal{F} \cup (\mathcal{I} \setminus \mathcal{V})$ (non-dropped indices)

$(r_d)_k^j = (c - A^\top y_k + Qx_k - z_k)^j$ dual infeasibility

Portfolio optimization: dropping & linear system solution

The optimal solution is expected to be (and actually is) sparse $\Longrightarrow$ dropping strategy:

- set a threshold $\epsilon_{\rm drop} > 0$ and a large constant $\xi > 0$
- iter $k = 0$: set $\mathcal{V} = \emptyset$
- iter $k > 0$: for every $j \in \mathcal{I} \setminus \mathcal{V}$, drop (i.e. set to 0) x_k^j and z_k^j such that

$$x_k^j \leq \epsilon_{\rm drop} \quad \text{and} \quad z_k^j \geq \xi \cdot \epsilon_{\rm drop} \quad \text{and} \quad (r_d)_k^j \leq \epsilon_{\rm drop}$$

and set $\mathcal{V} = \mathcal{V} \cup \{j\}$ (dropped indices), $\mathcal{G} = \mathcal{F} \cup (\mathcal{I} \setminus \mathcal{V})$ (non-dropped indices)

$(r_d)_k^j = (c - A^\top y_k + Q x_k - z_k)^j$ dual infeasibility

Solve by factorization the reduced augmented system corresponding to the non-dropped variables

$$\begin{bmatrix} -(\widehat{Q} + \widehat{\Xi}_k + \rho_k I) & \widehat{A}^\top \\ \widehat{A} & \delta_k I \end{bmatrix} \begin{bmatrix} \widehat{\Delta x} \\ \widehat{\Delta y} \end{bmatrix} = \begin{bmatrix} \widehat{r}_{1,k} \\ \widehat{r}_{2,k} \end{bmatrix}$$

much smaller system!

NOTE: a simple test at the end of the optimization process allows us to check if a variable was incorrectly dropped

Multi-period portfolio optimization: test setting

10 test problems generated from

- FF48-FF100 (Fama & French 48-100 Industry portfolios, USA), Jul 1926 – Dec 2015
- ES50 (EURO STOXX 50), 50 stocks from 9 Eurozone countries, Jan 2008 – Dec 2013
- FTSE100 (Financial Times Stock Exchange, UK), 100 assets, Jul 2002 – Apr 2016
- SP500 (Standard & Poors, USA), 500 assets, Jan 2008 – Dec 2016
- NASDAQC, almost all stocks in this stock market, Feb 2003 – Apr 2016

Comparison of IP-PMM with ASB-Chol (ad-hoc Alternating Split Bregman method)

MATLAB, implementation details in [De Simone, **dS**, Gondzio, Pougkiakiotis & Viola, to appear in SIAM Review 2022 (arXiv:2102.13608, 2021)

Performance metrics (comparison with multi-period naive portfolio)

- risk reduction factor: $ratio = \dfrac{w_{naive}^\top C\, w_{naive}}{w_{opt}^\top C\, w_{opt}}$
- holding cost reduction factor: $ratio_h = \dfrac{\#\text{ active positions of } w_{naive}}{\#\text{ active positions of } w_{opt}}$
- transaction reduction factor: $ratio_t = \dfrac{\mathcal{T}_{naive}}{\mathcal{T}_{opt}}$

$$\mathcal{T} = \text{ transaction cost } = trace(V^\top V), \quad v^{ij} = \begin{cases} 1 & \text{if } |w_j^i - w_{j+1}^i| \geq \epsilon = 10^{-4} \\ 0 & \text{otherwise} \end{cases}$$

Multi-period portfolio optimization: results

Problem ($\overline{n}$)	Time (s)	Iters	*ratio*	$ratio_h$	$ratio_t$
	IP-PMM				
FF48-10 (1632)	1.37e−1	12	2.32e+0	6.67e+0	1.66e+1
FF48-20 (3552)	3.77e−1	16	2.28e+0	6.58e+0	2.13e+1
FF48-30 (5472)	8.43e−1	21	4.64e+0	6.15e+0	1.69e+1
FF100-10 (3264)	4.92e−1	12	1.58e+0	1.78e+1	4.36e+1
FF100-20 (7104)	1.63e+0	15	1.81e+0	2.04e+1	4.92e+1
FF100-30 (10,944)	3.93e+0	21	5.82e+0	1.34e+1	3.60e+1
ES50 (4300)	4.59e−1	14	2.12e+0	4.42e+0	5.75e+1
FTSE100 (3154)	4.64e−1	14	1.85e+0	5.37e+1	6.09e+1
SP500 (11,206)	3.43e+1	16	1.57e+0	8.62e+1	1.50e+2
NASDAQC (45,714)	7.05e+2	20	3.15e+0	2.73e+0	3.89e+2
	ASB-Chol				
FF48-10 (1632)	1.67e−1	1431	2.33e+0	6.67e+0	1.66e+1
FF48-20 (3552)	3.72e−1	1985	2.31e+0	7.93e+0	2.09e+1
FF48-30 (5472)	1.12e+0	4125	4.64e+0	6.08e+0	1.66e+1
FF100-10 (3264)	8.49e−1	3087	1.58e+0	1.78e+1	4.36e+1
FF100-20 (7104)	2.09e+0	3635	1.80e+0	1.78e+1	4.27e+1
FF100-30 (10,944)	8.54e+0	9043	5.83e+0	1.12e+1	2.97e+1
ES50 (4300)	9.70e−1	4297	2.05e+0	2.94e+0	4.26e+1
FTSE100 (3154)	4.29e−1	1749	1.80e+0	5.07e+1	5.71e+1
SP500 (11,206)	1.98e+1	3728	1.74e+0	6.16e+1	1.01e+2
NASDAQC (45,714)	8.84e+2	14264	3.15e+0	2.73e+0	3.89e+2

Application 2: binary classification of fMRI data

- $s_{(-1)}$ 3d scans in class "-1" and $s_{(1)}$ 3d scans in class "1", $s = s_{(-1)} + s_{(1)}$
- Each 3d scan is a $q_1 \times q_2 \times q_3$ real array ($q = q_1 q_2 q_3$ voxels)
- $D \in \mathbb{R}^{s \times q}$ matrix containing as rows the 3d scans (reshaped as vectors)
- $\hat{y}$ vector containing the labels associated with each scan
- Square loss function for determining a separating hyperplane in $\mathbb{R}^q$
- $\#$ patients much smaller than the scan size i.e. $s \ll q$ (ill-posed problem), similar weights of the classification hyperplane sought for contiguous brain regions ("structured" sparsity)
  $\Longrightarrow$ regularization with ℓ_1 and anisotropic Total Variation (TV) terms

Application 2: binary classification of fMRI data

- $s_{(-1)}$ 3d scans in class "-1" and $s_{(1)}$ 3d scans in class "1", $s = s_{(-1)} + s_{(1)}$
- Each 3d scan is a $q_1 \times q_2 \times q_3$ real array ($q = q_1 q_2 q_3$ voxels)
- $D \in \mathbb{R}^{s \times q}$ matrix containing as rows the 3d scans (reshaped as vectors)
- $\hat{y}$ vector containing the labels associated with each scan
- Square loss function for determining a separating hyperplane in $\mathbb{R}^q$
- # patients much smaller than the scan size i.e. $s \ll q$ (ill-posed problem), similar weights of the classification hyperplane sought for contiguous brain regions ("structured" sparsity)
  $\Longrightarrow$ regularization with ℓ_1 and anisotropic Total Variation (TV) terms

$$\min_w \; \frac{1}{2s} \|Dw - \hat{y}\|^2 + \tau_1 \|w\|_1 + \tau_2 \|Lw\|_1$$

$\tau_1, \tau_2 > 0$, $\quad \|Lw\|_1$ discrete anisotropic TV of w

$L = [L_x^\top \;\; L_y^\top \;\; L_z^\top]^\top \in \mathbb{R}^{l \times q}$ first-order forward finite differences in x, y, z

[Baldassarre, Pontil & Mouraõ-Miranda, Front Neurosci 2017]

Application 2: binary classification of fMRI data (cont'd)

Smooth problem reformulation

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^\top Q x + \; c^\top x, \\ \text{s.t.} \quad & Ax = b, \;\; x_{\mathcal{I}} \geq 0, \; x_{\mathcal{F}} \text{ free}, \; \mathcal{I} = \{s+1, \ldots, n\}, \; \mathcal{F} = \{1, \ldots, s\}, \end{aligned}$$

$$u = Dw, \quad d = Lw, \quad w = w^+ - w^-, \; d = d^+ - d^-$$

$$x = [u^\top, (w^+)^\top, (w^-)^\top, (d^+)^\top, (d^-)^\top]^\top$$

$$Q = \begin{bmatrix} \frac{1}{s} I_s & 0_{s,(n-s)} \\ 0_{(n-s),s} & 0_{(n-s),(n-s)} \end{bmatrix}, \quad A = \begin{bmatrix} -I_s & D & -D & 0_{s,l} & 0_{s,l} \\ 0_{l,s} & L & -L & -I_l & I_l \end{bmatrix}$$

(diagonal Hessian)

$$c = \left[-\frac{\hat{y}^\top}{s}, \tau_1 e_w^\top, \tau_1 e_w^\top, \tau_2 e_d^\top, \tau_2 e_d^\top \right]^\top \in \mathbb{R}^n, \quad b = 0_{s+l} \in \mathbb{R}^m, \quad m = l+s, \quad n = s+2q+2l$$

Classification of fMRI data: solution of Newton system

- $\nabla^2 f(x_k) = Q$ diagonal $\implies$ solve the (spd) normal equations:

$$M_k \Delta y = r, \quad M_k = A(Q + \Xi_k + \rho_k I_n)^{-1} A^\top + \delta_k I_m$$

- $M_k = \begin{bmatrix} M_{1,k} & M_{2,k}^\top \\ M_{2,k} & M_{3,k} \end{bmatrix}$ $\quad$ $M_{1,k}, M_{2,k}$ dense
  $M_{3,k}$ sparse, size $l \gg s$

  $\implies$ use Preconditioned Conjugate Gradient (PCG) method

- Preconditioner:

$$P_k = \begin{bmatrix} M_{1,k} & 0 \\ 0 & M_{3,k} \end{bmatrix} \quad \text{block diagonal}$$

  $M_{3,k}$ has a sparse Cholesky factor (thanks to TV matrix L)

  $M_{1,k}$ has a dense Cholesky factor, requiring only $O(s^3)$ operations and $O(s^2)$ storage

Classification of fMRI data: spectral analysis

Theorem

The preconditioned matrix $R_k = P_k^{-1} M_k$ *has* $l - \mathrm{rank}(D)$ *eigenvalues* $\lambda = 1$, *whose respective eigenvectors form a basis for* $\{0_s\} \times \{Null(M_{2,k}^\top)\}$. *All the remaining eigenvalues of the preconditioned matrix satisfy*

$$\lambda \in (\chi, 1) \cup (1, 2), \quad \chi = \frac{\delta_k \rho_k}{\sigma_{\max}^2(A) + \rho_k \delta_k},$$

where δ_k, ρ_k *are the regularization parameters of IP-PMM.*

The preconditioner remains effective as long as ρ_k and δ_k are not too small

$\mathcal{A} \times \mathcal{B}$ denotes a vector space with elements $[a^\top, b^\top]^\top$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$

Classification of fMRI data: dropping strategy

- ρ_k and δ_k must be reduced to attain convergence of IP-PMM
- the optimal solution is expected to be sparse
  $\Longrightarrow$ drop primal variables converging to 0 to improve matrix conditioning (same strategy as in the portfolio problem)

Reduced normal equations

$$\left(\widetilde{A}\left(\widetilde{Q}+\widetilde{\Xi}_k+\rho_k I\right)^{-1}\widehat{A}^{\top}+\delta_k I\right)\widehat{\Delta y}=\widehat{r}$$

smaller and "safer" system!

Classification of fMRI data: test setting

(Preprocessed) data from https://github.com/lucabaldassarre/neurosparse

- fMRI scans for 16 male healthy US college students (age 20 to 25), two active conditions: viewing unpleasant and pleasant images
- 1344 scans of size 122,128 voxels (only voxels with probability > 0.5 of being in the gray matter), 42 scans per subject and active condition (i.e., 84 scans per subject in total)
- Leave-One-Subject-Out (LOSO) cross-validation test over the full dataset of patients
  $\Longrightarrow$ size of w: $q = 122{,}128$, $\#$ rows D: $s = 1260$, size of $d = Lw$: $l = 339{,}553$

Comparison of IP-PMM with ad-hoc FISTA and ADMM

MATLAB, implementation details in [De Simone, **dS**, Gondzio, Pougkakiotis & Viola, to appear in SIAM Review 2022 (arXiv:2102.13608, 2021)]

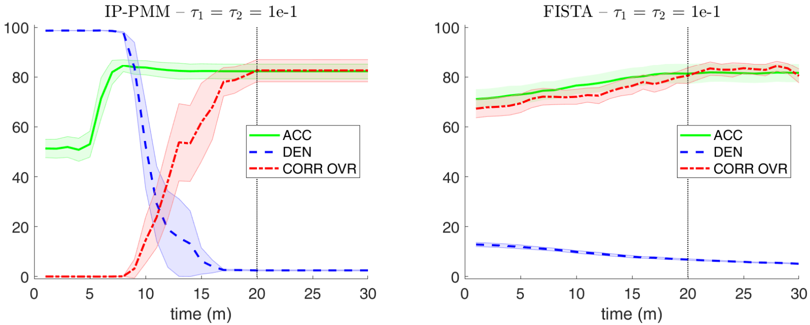
Performance metrics [Baldassarre, Pontil & Mouraõ-Miranda, Front Neurosci 2017]

- classification accuracy (ACC): percentage of vectors correctly classified
- solution density (DEN): percentage of nonzero entries
- corrected pairwise overlap (CORR OVR): measure of “stability” of the voxel selection, the higher the better

Classification of fMRI data: results

Algorithm	$\tau_1 = \tau_2$	ACC	DEN	CORR OVR
IP-PMM	10^{-2}	86.16 ± 7.11	20.56 ± 6.63	43.47 ± 9.09
	$5 \cdot 10^{-2}$	84.90 ± 4.80	3.77 ± 0.84	62.70 ± 10.39
	10^{-1}	82.29 ± 6.22	2.49 ± 0.34	82.60 ± 9.24
FISTA	10^{-2}	86.90 ± 5.01	88.97 ± 0.71	5.43 ± 0.43
	$5 \cdot 10^{-2}$	84.15 ± 5.92	19.36 ± 0.86	65.50 ± 2.68
	10^{-1}	81.62 ± 7.58	5.14 ± 0.44	80.44 ± 5.72
ADMM	10^{-2}	86.46 ± 6.91	98.70 ± 0.03	0.03 ± 0.01
	$5 \cdot 10^{-2}$	85.57 ± 5.37	97.97 ± 0.05	0.15 ± 0.04
	10^{-1}	82.07 ± 6.51	97.50 ± 0.19	0.26 ± 0.13

Classification of fMRI data: results (cont'd)



Application 3: TV-based Poisson image restoration

- Object to be restored: $w \in \mathbb{R}^n$, measured data: $g \in \mathbb{N}_0^m$, with entries g^j that are samples of m independent random variables $G^j \sim Poisson((Dw + a)^j)$
- $D = [d^{ij}] \in \mathbb{R}^{m \times n}$ modeling the imaging system, $d^{ij} \geq 0$ for all i, j, $\sum_{i=1}^m d^{ij} = 1$ for all j, BCCB structure assumed
- $a \in \mathbb{R}_+^m$ modeling the background radiation detected by the sensors
- Maximum-likelihood approach $\Longrightarrow$ minimization of Kullback-Leibler (KL) divergence (highly ill-conditioned problem) $\Longrightarrow$ TV regularization
- Non-negative image intensity, total image intensity preserved $\Longrightarrow$ non-negativity + single linear constraint

Application 3: TV-based Poisson image restoration

- Object to be restored: $w \in \mathbb{R}^n$, measured data: $g \in \mathbb{N}_0^m$, with entries g^j that are samples of m independent random variables $G^j \sim Poisson((Dw + a)^j)$
- $D = [d^{ij}] \in \mathbb{R}^{m \times n}$ modeling the imaging system, $d^{ij} \geq 0$ for all i, j, $\sum_{i=1}^m d^{ij} = 1$ for all j, BCCB structure assumed
- $a \in \mathbb{R}_+^m$ modeling the background radiation detected by the sensors
- Maximum-likelihood approach $\Longrightarrow$ minimization of Kullback-Leibler (KL) divergence (highly ill-conditioned problem) $\Longrightarrow$ TV regularization
- Non-negative image intensity, total image intensity preserved $\Longrightarrow$ non-negativity + single linear constraint

$$\min_w \; D_{KL}(w) + \lambda \|Lw\|_1$$
$$\text{s.t.} \; e_n^\top w = r, \quad w \geq 0$$

$$D_{KL}(w) = \sum_{j=1}^m \left(g^j \ln \frac{g^j}{(Dw+a)^j} + (Dw + a)^j - g^j \right)$$
$$L \in \mathbb{R}^{l \times n} \text{ discrete TV operator}, \;\; r = \sum_{j=1}^m (g^j - a^j)$$

Appl. 3: TV-based Poisson image restoration (cont'd)

Smooth problem reformulation

$$\begin{array}{ll} \min\limits_{x} & f(x) \equiv D_{KL}(w) + c^\top u, \\ \text{s.t.} & Ax = b, \;\; x \geq 0 \end{array}$$

$$d = Lw, \quad u = [(d^+)^\top,\ (d^-)^\top]^\top, \quad x = [w^\top,\ u^\top]^\top$$

$$A = \begin{bmatrix} e_n^\top & 0_l^\top & 0_l^\top \\ L & -I_l & I_l \end{bmatrix}$$

$$c = \lambda\, e_{2l}, \quad b = [r,\ 0_l^\top]^\top \in \mathbb{R}^{\overline{m}}, \quad \overline{m} = l+1, \quad \overline{n} = n+2l, \quad m = l+s, \quad n = s+2q+2l$$

TV-based Poisson image restoration: Newton system

- $$\underbrace{\begin{bmatrix} -H_k & A^\top \\ A & \delta_k I \end{bmatrix}}_{M_k} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{1,k} \\ r_{2,k} \end{bmatrix}, \qquad H_k = (\nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I)$$

  $\Longrightarrow$ use preconditioned MINimum RESidual (MINRES) method

- Preconditioner:

$$\widetilde{M}_k = \begin{bmatrix} \widetilde{H}_k & 0 \\ 0 & A\,\widetilde{H}_k^{-1} A^\top + \delta_k I \end{bmatrix}, \qquad \widetilde{H}_k \text{ diagonal approx of } H_k$$

Theorem

The eigenvalues of $\widetilde{M}_k^{-1} M_k$ lie in the union of the intervals

$$I_- = \left[-\beta_H - 1, -\alpha_H \right], \qquad I_+ = \left[\frac{1}{1+\beta_H}, 1 \right],$$

where $\alpha_H = \lambda_{\min}(\widehat{H}_k)$, $\beta_H = \lambda_{\max}(\widehat{H}_k)$ and $\widehat{H}_k = \widetilde{H}_k^{-\frac{1}{2}} H_k \widetilde{H}_k^{\frac{1}{2}}$.

[Bergamaschi, Gondzio, Martínez, Pearson & Pougkakiotis, NLAA 2021]

If $\widetilde{H}_k = \mathrm{diag}(H_k)$, then $\alpha_H \leq 1 \leq \beta_H$

TV-based Poisson image restoration: Newton sys (cont'd)

- $$\begin{bmatrix} -H_k & A^\top \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{1,k} \\ r_{2,k} \end{bmatrix}, \qquad H_k = (\nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I)$$

- $$\nabla^2 f(x) = \begin{bmatrix} \nabla^2 D_{KL}(w) & 0 \\ 0 & 0 \end{bmatrix}, \qquad \nabla^2 D_{KL}(w) = D^\top U(w)^2 D$$

  $$U(w) = \mathrm{diag}\left(\frac{\sqrt{g}}{Dw + a} \right), \qquad \widetilde{H}_k = U(w_k)^2$$

  D may be dense, but its action of a vector can be computed via FFT

  $\widetilde{H}_k = U(w_k)^2$ better than $\widetilde{H}_k = \mathrm{diag}(H_k)$

TV-based Poisson image restoration: test setting

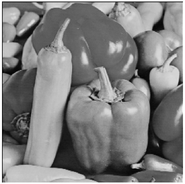
Test images

- 256×256, grayscale

cameraman

house

peppers



- Poisson noise and Gaussian blur (GB), motion blur (MB), out-of-focus blur (OF)

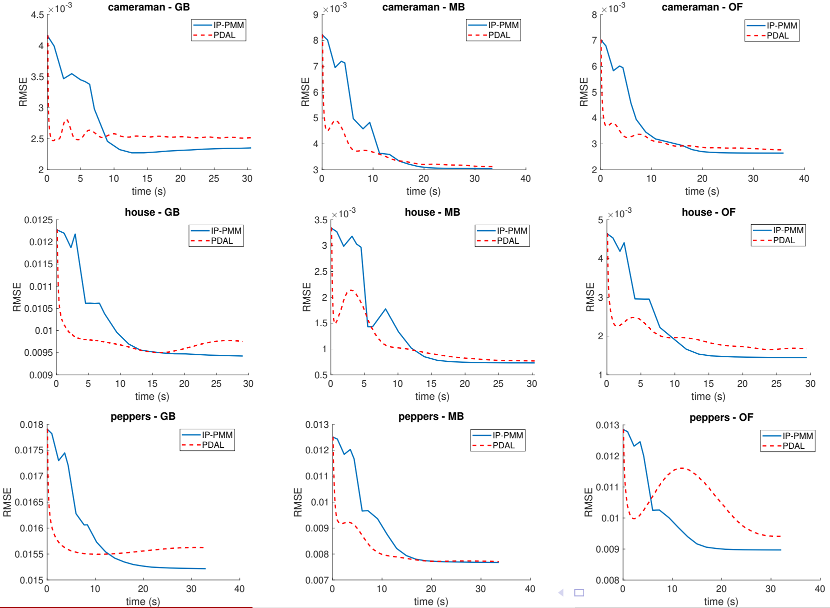
Comparison of IP-PMM with Primal-Dual Algorithm with Linesearch (PDAL)

MATLAB, implementation details in [De Simone, **dS**, Gondzio, Pougkakiotis & Viola, to appear in SIAM Review 2022 (arXiv:2102.13608, 2021)]

Performance metrics

- $\mathrm{RMSE}(w) = \frac{1}{\sqrt{n}}\|w - \overline{w}\|_2$, $\overline{w}$ original image
- $\mathrm{PSNR}(w) = 20 \log_{10}(\max_i \overline{w}^i / \mathrm{RMSE}(w))$
- MSSIM = structural similarity measure, the higher the better

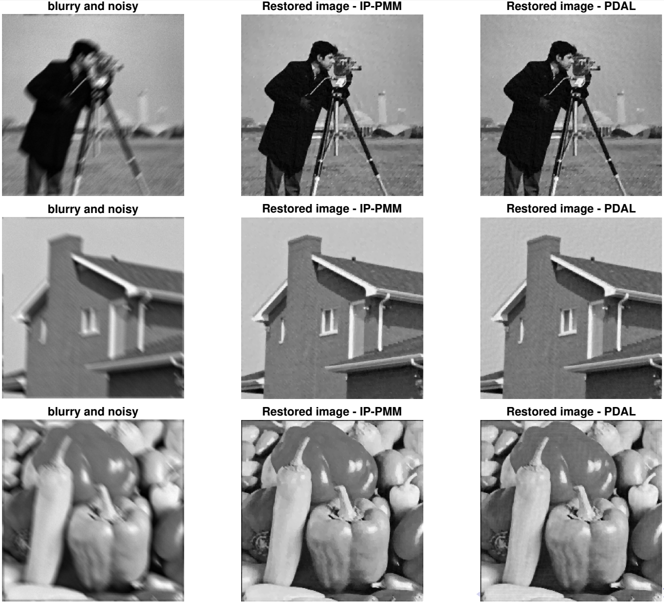
TV-based Poisson image restoration: results



TV-based Poisson image restoration: results (cont'd)

	IP-PMM			PDAL		
Problem	**RMSE**	**PSNR**	**MSSIM**	**RMSE**	**PSNR**	**MSSIM**
cameraman - GB	4.85e−2	2.63e+1	8.33e−1	5.02e−2	2.60e+1	8.22e−1
cameraman - MB	5.52e−2	2.52e+1	8.11e−1	5.59e−2	2.51e+1	7.77e−1
cameraman - OF	5.14e−2	2.58e+1	7.98e−1	5.26e−2	2.56e+1	7.62e−1
house - GB	9.71e−2	2.03e+1	7.51e−1	9.88e−2	2.01e+1	6.92e−1
house - MB	2.70e−2	3.14e+1	8.67e−1	2.77e−2	3.11e+1	8.43e−1
house - OF	3.80e−2	2.84e+1	8.33e−1	4.09e−2	2.78e+1	7.70e−1
peppers - GB	1.23e−1	1.82e+1	7.46e−1	1.25e−1	1.81e+1	6.57e−1
peppers - MB	8.76e−2	2.12e+1	8.90e−1	8.78e−2	2.11e+1	8.72e−1
peppers - OF	9.47e−2	2.05e+1	8.01e−1	9.70e−2	2.03e+1	6.60e−1

TV-based Poisson image restoration: results (cont'd)



Appl. 4: linear classification via Logistic Regression

- Training set with n binary-labeled samples and s features
- $D \in \mathbb{R}^{n \times s}$ with rows $(d^i)^\top$ representing the training points
- $g \in \{-1, 1\}^n$ vector of labels
- Logistic model to define the conditional probability of having the label g^i given the point d^i
- Maximum-likelihood approach $\Longrightarrow$ minimization of logistic loss function (ill posedness – e.g. redundant features) $\Longrightarrow$ ℓ_1 regularization

Appl. 4: linear classification via Logistic Regression

- Training set with n binary-labeled samples and s features
- $D \in \mathbb{R}^{n \times s}$ with rows $(d^i)^\top$ representing the training points
- $g \in \{-1, 1\}^n$ vector of labels
- Logistic model to define the conditional probability of having the label g^i given the point d^i
- Maximum-likelihood approach $\implies$ minimization of logistic loss function (ill posedness – e.g. redundant features) $\implies$ ℓ_1 regularization

$$\min_w \; \phi(w) + \tau \|w\|_1$$

$$\phi(w) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(w), \quad \phi_i(w) = \log\left(1 + e^{-g^i\, w^\top d^i}\right)$$

Appl. 4: linear classification via Logistic Regression (cont'd)

Smooth problem reformulation

$$\min_{x} \; f(x) \equiv \phi(w) + c^\top u$$
$$\text{s.t. } Ax = b, \;\; u \geq 0$$

$$u = w, \quad u = [(d^+)^\top, \, (d^-)^\top]^\top, \quad x = [w^\top, \; u^\top]^\top$$

$$A = [I_s \quad -I_s \quad I_s]$$

$$c = \tau \, e_{2s}, \quad b = 0_{\overline{m}}, \quad \overline{m} = l + 1, \quad \overline{m} = s, \quad \overline{n} = 3s$$

Classific. via Logistic Regression: solution of Newton system

- Solution of Newton system by preconditioned MINRES (similar to Poisson image restoration)

- Preconditioner:

$$\widetilde{M}_k = \begin{bmatrix} \widetilde{H}_k & 0 \\ 0 & A\,\widetilde{H}_k^{-1} A^\top + \delta_k I \end{bmatrix}$$

$$\widetilde{H}_k = \mathrm{diag}(H_k), \quad H_k = (\nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I)$$

Classification via Logistic Regression: test setting

Linear classification problems from the LIBSVM dataset for binary classification, https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

Problem	Features	Train pts	Test pts
gisette	5000	6000	1000
rcv1	47,236	20,242	677,399
real-sim	20,958	50,617	21,692

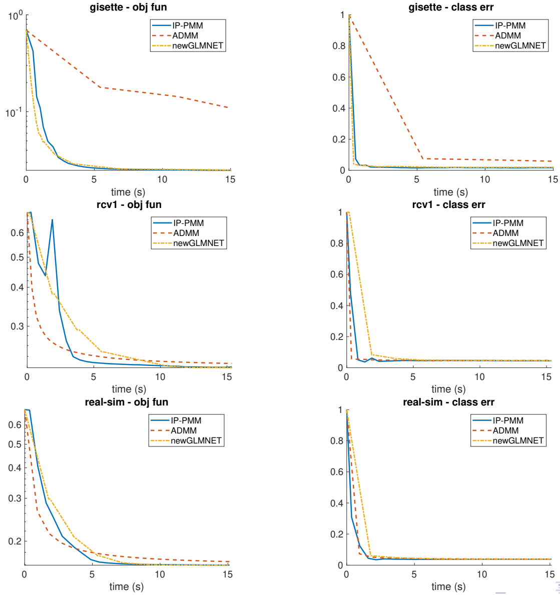
Comparison of IP-PMM with ADMM (http://www.stanford.edu/~boyd/papers/distr_opt_stat_learning_admm.html) and newGLMNET used in LIBSVM (https://github.com/ZiruiZhou/IRPN)

MATLAB, implementation details in [De Simone, **dS**, Gondzio, Pougkakiotis & Viola, to appear in SIAM Review 2022 (arXiv:2102.13608, 2021)]

Performance metrics

- objective function value versus execution time
- classification error versus execution time

Classification via Logistic Regression: results



Conclusions

- Specialized IPMs for quadratic and general convex nonlinear optimization problems with sparse solutions have been presented
- By a proper choice of linear algebra solvers, IPMs can efficiently solve larger but smooth optimization problems coming from a standard reformulation of the original ones
- Computational experiments on diverse applications provide evidence that IPMs can offer a noticeable advantage over state-of-the-art first-order methods, especially when dealing with not-so-well conditioned problems
- This work may provide a basis for an in-depth analysis of the application of IPMs to many sparse approximation problems

Some references

- V. De Simone, **dS**, J. Gondzio, S. Pougkakiotis, M. Viola, *Sparse Approximations with Interior Point Methods*, to appear in SIAM Review 2022 (arXiv:2102.13608, 2021)
- S. Cafieri, M. D'Apuzzo, V. De Simone, **dS**, *On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems*, Computational Optimization and Applications, 38 (2007)
- S. Corsaro, V. De Simone, Z. Marino, *Fused lasso approach in portfolio selection*, Annals of Operations Research (2019)
- M. D'Apuzzo, V. De Simone, D. di Serafino, *On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods*, Computational Optimization and Applications, 45 (2010)
- **dS**, G. Landi, M. Viola, *ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration* Applied Mathematics and Computation, 364 (2020)
- J. Gondzio, *Interior point methods 25 years later*, European Journal of Operational Research, 218 (2012)
- S. Pougkakiotis, J. Gondzio, *An interior point-proximal method of multipliers for convex quadratic programming*, Computational Optimization and Applications, 78 (2021)

Thank you for your attention!