Adaptive Subspace Methods for Optimization

Daniel P. Robinson
Johns Hopkins University
Department of Applied Mathematics and Statistics

Collaborators:
Tianyi Chen (PhD student at Johns Hopkins University)
Frank E. Curtis (Lehigh University)
Hassan Mohy-ud-Din (Postdoc student at Yale School of Medicine)

Funding:
IDIES Seed Funding Program
National Science Foundation (DMS-1217153)

U.S.-Mexico Workshop on Optimization and its Applications
Merida, Mexico
January 4, 2016

It all started with Jorge:

- As a postdoc, designed first two-phase algorithm for asymmetric LCP.
- Our method was good for a strictly convex bound-constrained problem (BQP).
- Learned that matrix-splitting iterations were better at active-set identification than simple gradient projection iterations (think Moré and Toraldo).
- Referee directed us to a work by Dostál and Schöberl for strictly convex BQP.
- Our algorithm was better on well-conditioned problems, but performance deteriorated with the condition number.

Our method:

- Active-set prediction via matrix-splittings (e.g., projected gradient).
- Terminate CG when residual is fraction of initial residual or max iterations.

Dostál and Schöberl (and Diniz-Ehrhardt, Friedlander, Martinez):

- Terminate CG when (opt. error in free-space) < (opt. error in active-space)
  - example of Adaptive Subspace Minimization (ASM) condition.
- When CG is terminated, use gradient in active-space to make free-space larger.
  - example of evolving subspace using a Subspace Restricted Projection (SRP).
- Theory:
  - finite termination even when problems are dual degenerate.
  - once the iterates leave a subspace, they will never return.
- Numerics:
  - Very efficient.
  - Robust to condition number of Hessian matrix.

Our method:

- Active-set prediction via matrix-splittings (e.g., projected gradient).
- Terminate CG when residual is fraction of initial residual or max iterations.

Dostál and Schöberl (and Diniz-Ehrhardt, Friedlander, Martinez):

- Terminate CG when (opt. error in free-space) < (opt. error in active-space)
  - example of Adaptive Subspace Minimization (ASM) condition.
- When CG is terminated, use gradient in active-space to make free-space larger.
  - example of evolving subspace using a Subspace Restricted Projection (SRP).
- Theory:
  - finite termination even when problems are dual degenerate.
  - once the iterates leave a subspace, they will never return.
- Numerics:
  - Very efficient.
  - Robust to condition number of Hessian matrix.

How useful are these ideas for more general problems?

Outline

1. Nonconvex BQP
   - Our Algorithm
   - Numerical Results
   - Summary and Future Work
2. Sparse Nonlinear Convex Optimization
   - Our Algorithm
   - Numerical Results
   - Summary and Future Work

Outline

Outline

The target problem.

We want to solve the nonconvex bound-constrained quadratic program (BQP):

$$\underset{x}{\text{minimize}} \;\; q(x) := \tfrac{1}{2} x^T H x - c^T x \;\; \text{subject to} \;\; \ell \leq x \leq u$$

- $c \in \mathbb{R}^n$
- $H \in \mathbb{R}^{n \times n}$ (not necessarily positive semidefinite)
- $g(x) := \nabla q(x) = Hx - c$
- $\ell < u$ (infinite bounds are allowed)

Algorithm 1 Algorithm for a nonconvex BQP.

1: **loop**
2: **if** (error in free-space) > (error in active-space) **then** ▷ ASM
3: Compute CG step s_k.
4: **if** $s_k^T H s_k > 0$ **then**
5: Compute CG step length α_{cg} and longest feasible step length α_{feas}.
6: **if** $\alpha_{cg} \leq \alpha_{\text{feas}}$ **then**
7: Accept the CG iteration: $x_{k+1} \leftarrow x_k + \alpha_{cg} s_k$.
8: **else**
9: Take step to boundary: $\widehat{x}_k \leftarrow x_k + \alpha_{feas} s_k$.
10: Set x_{k+1} as a gradient projection in active-space at $\widehat{x}_k$. ▷ SRP
11: **else**
12: Do computations in lines 9–10.
13: **else**
14: Compute a step that frees-up some of the active-variables. ▷ SRP

Definition (active set, free set, lower-active, and upper-active sets)

$$\mathcal{A}(x) := \{i : [x]_i \in \{\ell_i, u_i\}\} \qquad \mathcal{F}(x) := \{1, 2, \ldots, n\} \setminus \mathcal{A}(x)$$

$$\mathcal{A}_l(x) := \{i \in \mathcal{A}(x) : [x]_i = \ell_i\} \qquad \mathcal{A}_u(x) := \{i \in \mathcal{A}(x) : [x]_i = u_i\}$$

Definition (active set, free set, lower-active, and upper-active sets)

$$\mathcal{A}(x) := \{i : [x]_i \in \{\ell_i, u_i\}\} \qquad \mathcal{F}(x) := \{1, 2, \ldots, n\} \setminus \mathcal{A}(x)$$
$$\mathcal{A}_l(x) := \{i \in \mathcal{A}(x) : [x]_i = \ell_i\} \qquad \mathcal{A}_u(x) := \{i \in \mathcal{A}(x) : [x]_i = u_i\}$$

Definition (reduced free-gradient and active-gradient)

$$\varphi(x) := \begin{cases} \min\{x - l, g(x)\} & \text{if } -g(x) \leq 0 \text{ and in } \mathcal{F}(x) \\ \max\{x - u, g(x)\} & \text{if } -g(x) > 0 \text{ and in } \mathcal{F}(x) \\ 0 & \text{on } \mathcal{A}(x) \end{cases}$$

and

$$\beta(x) := \begin{cases} \min\{x - l, [g(x)]^+\} & \text{on } \mathcal{A}_u(x) \\ \max\{x - u, [g(x)]^-\} & \text{on } \mathcal{A}_l(x) \\ 0 & \text{on } \mathcal{F}(x) \end{cases}$$

Definition (active set, free set, lower-active, and upper-active sets)

$$\mathcal{A}(x) := \{i : [x]_i \in \{\ell_i, u_i\}\} \qquad \mathcal{F}(x) := \{1, 2, \ldots, n\} \setminus \mathcal{A}(x)$$
$$\mathcal{A}_l(x) := \{i \in \mathcal{A}(x) : [x]_i = \ell_i\} \qquad \mathcal{A}_u(x) := \{i \in \mathcal{A}(x) : [x]_i = u_i\}$$

Definition (reduced free-gradient and active-gradient)

$$\varphi(x) := \begin{cases} \min\{x - l, g(x)\} & \text{if } -g(x) \leq 0 \text{ and in } \mathcal{F}(x) \\ \max\{x - u, g(x)\} & \text{if } -g(x) > 0 \text{ and in } \mathcal{F}(x) \\ 0 & \text{on } \mathcal{A}(x) \end{cases}$$

and

$$\beta(x) := \begin{cases} \min\{x - l, [g(x)]^+\} & \text{on } \mathcal{A}_u(x) \\ \max\{x - u, [g(x)]^-\} & \text{on } \mathcal{A}_l(x) \\ 0 & \text{on } \mathcal{F}(x) \end{cases}$$

Comments:

- Let P_Ω be the projection operator onto the feasible region Ω.
- $P_\Omega(x - g(x)) = x - \big(\varphi(x) + \beta(x)\big)$.
- Optimality measure in free-variables: $g(x)^T\varphi(x)$
- Optimality measure in active-variables: $g(x)^T\beta(x)$

Algorithm 2 Algorithm for a nonconvex BQP.

1: **loop**
2: **if** $g(x_k)^T\varphi(x_k) > g(x_k)^T\beta(x_k)$ **then** ▷ ASM
3: Compute CG step s_k.
4: **if** $s_k^T H s_k > 0$ **then**
5: Compute CG step length α_{cg} and longest feasible step length α_{feas}.
6: **if** $\alpha_{cg} \leq \alpha_{\text{feas}}$ **then**
7: Accept the CG iteration: $x_{k+1} \leftarrow x_k + \alpha_{cg} s_k$.
8: **else**
9: Take step to boundary: $\widehat{x}_k \leftarrow x_k + \alpha_{feas} s_k$.
10: Set $x_{k+1} \leftarrow P_\Omega(\widehat{x}_k - \varphi(\widehat{x}_k))$ for some α_k. ▷ SRP
11: **else**
12: Do computations in lines 9–10.
13: **else**
14: Set $x_{k+1} \leftarrow x_k - \alpha_k \beta(x_k)$ for some α_k. ▷ SRP

Outline

Convex problems

		PGM			Our Method		
			Hx Prods.			*Hx* Prods.	
n	H_{cond}	Solved	mean	s.d.	Solved	mean	s.d.
1e+02	1e+02	100%	104.0	12.7	100%	54.6	6.4
1e+02	1e+04	100%	525.0	478.0	100%	125.0	21.0
1e+02	1e+06	100%	645.0	591.0	100%	132.0	38.3
1e+02	1e+08	100%	874.0	1710.0	100%	100.0	22.4
1e+03	1e+02	100%	230.0	45.0	100%	111.0	5.9
1e+03	1e+04	100%	4640.0	4420.0	100%	415.0	29.7
1e+03	1e+06	84%	29500.0	24300.0	100%	1230.0	168.0
1e+03	1e+08	60%	44600.0	35000.0	100%	1300.0	173.0

Table : Comparison with PGM on 50 randomly generated strictly convex problems.

Nonconvex problems

		PGM			Our Method		
			Hx Prods.			*Hx* Prods.	
n	H_{cond}	Solved	mean	s.d.	Solved	mean	s.d.
1e+02	1e+02	100%	139.0	208.0	100%	85.7	9.2
1e+02	1e+04	98%	4510.0	6570.0	100%	145.0	23.6
1e+02	1e+06	94%	19600.0	19000.0	100%	165.0	44.4
1e+02	1e+08	92%	20500.0	24200.0	100%	131.0	34.8
1e+03	1e+02	100%	28800.0	13800.0	100%	328.0	30.9
1e+03	1e+04	98%	11500.0	11400.0	100%	744.0	70.8
1e+03	1e+06	98%	69800.0	19700.0	100%	1540.0	262.0
1e+03	1e+08	50%	65700.0	21000.0	100%	1640.0	237.0

Table : Comparison with PGM on 50 randomly generated nonconvex problems.

Outline

Summary:

- Key changes:
  - negative curvature CG steps
  - saddle point steps
  - reduced active-gradient
- Preliminary numerical results are promising.
- Method is less sensitive to the condition number (as much as can be expected) when compared to projected gradient method.

Future Work:

- Preconditioning is possible in both free-space and active-space computations.
- Include the new solver within an augmented Lagrangian framework.
- Extend the idea of a Cauchy point to allow inexact solves.

Outline

Outline

The optimization problem.

$$\min_{x \in \mathbb{R}^n} \; f(x) + \lambda \|x\|_1$$

- f is twice-continuously differentiable, $\nabla^2 f(x) \succ 0$, and $\lambda > 0$

Different approaches

- Convert to an equivalent differentiable problem

$$\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^n} \; f(u - v) + \lambda e^T (u + v) \;\text{ subject to }\; (u, v) \geq 0$$

  and then solve using first- or second-order projection methods.
- Deal directly with the non-differentiability
  - first-order : steepest descent (minimum norm element of subdifferential)
    ISTA/FISTA (Beck and Teboulle)
    SpaRSA (Wright, Nowak, and Figueiredo)
  - second-order : proximal Newton : LIBLINEAR (coordinate descent)
    orthant-based : OBA (Keskar, Nocedal, Öztoprak, and Wächter)
    OWL-QN (Andrew and Gao)

(u, v) formulation

+

extend ideas we recently developed for bound-constrained QP

+

standard projections onto non-negative orthant

+

convert back to x-space

=

new algorithm

(u, v) formulation

+

extend ideas we recently developed for bound-constrained QP

+

standard projections onto non-negative orthant

+

convert back to x-space

=

new algorithm

(orthant-ideas + modified projections + reduced ISTA step)

Algorithm 3 A method for sparse convex optimization.

Set $k \leftarrow 0$ and choose $\{\eta_\phi, \eta_\beta\} \subset (0,1)$.
loop
  Compute $\mathcal{I}^0(x_k) = \{i : [x_k]_i = 0\}$, $\mathcal{I}^+(x_k) = \{i : [x_k]_i > 0\}$, $\mathcal{I}^-(x_k) = \{i : [x_k]_i < 0\}$.
  if $\|\phi(x_k)\| \geq \|\beta(x_k)\|$ **then** ▷ ASM
    Choose any $\mathcal{I}_k \subseteq \{i : [\phi(x_k)]_i \neq 0\}$ such that $\|[\phi(x_k)]_{\mathcal{I}_k}\| \geq \eta_\phi \|\phi(x_k)\|$.
    Set $H_k \leftarrow [\nabla^2 F(x_k)]_{\mathcal{I}_k\mathcal{I}_k}$ and $g_k \leftarrow [\nabla F(x_k)]_{\mathcal{I}_k}$.
    Compute the reference direction $d_k^R \leftarrow -\alpha_k g_k$, where $\alpha_k \leftarrow \|g_k\|^2/(g_k^T H_k g_k)$.
    Define the model $m_k(d) = g_k^T d + \frac{1}{2} d^T H_k d$.
    Compute any direction $\bar{d}_k$ that satisfies the inequalities

$$g_k^T \bar{d}_k \leq g_k^T d_k^R \quad \text{and} \quad m_k(\bar{d}_k) \leq m_k(0).$$

    Set $[d_k]_{\mathcal{I}_k} \leftarrow \bar{d}_k$ and $[d_k]_i \leftarrow 0$ for $i \notin \mathcal{I}_k$.
    Compute $x_{k+1} = \text{LINESEARCH\_}\phi(x_k, d_k)$. ▷ SRP
  else
    Choose any $\mathcal{I}_k \subseteq \{i : [\beta(x_k)]_i \neq 0\}$ such that $\|[\beta(x_k)]_{\mathcal{I}_k}\| \geq \eta_\beta \|[\beta(x_k)]\|$.
    Set $[d_k]_{\mathcal{I}_k} \leftarrow -[\beta(x_k)]_{\mathcal{I}_k}$ and $[d_k]_i \leftarrow 0$ for $i \notin \mathcal{I}_k$.
    Compute $x_{k+1} = \text{LINESEARCH\_}\beta(x_k, d_k)$. ▷ SRP
  Set $k \leftarrow k + 1$.

Comments:

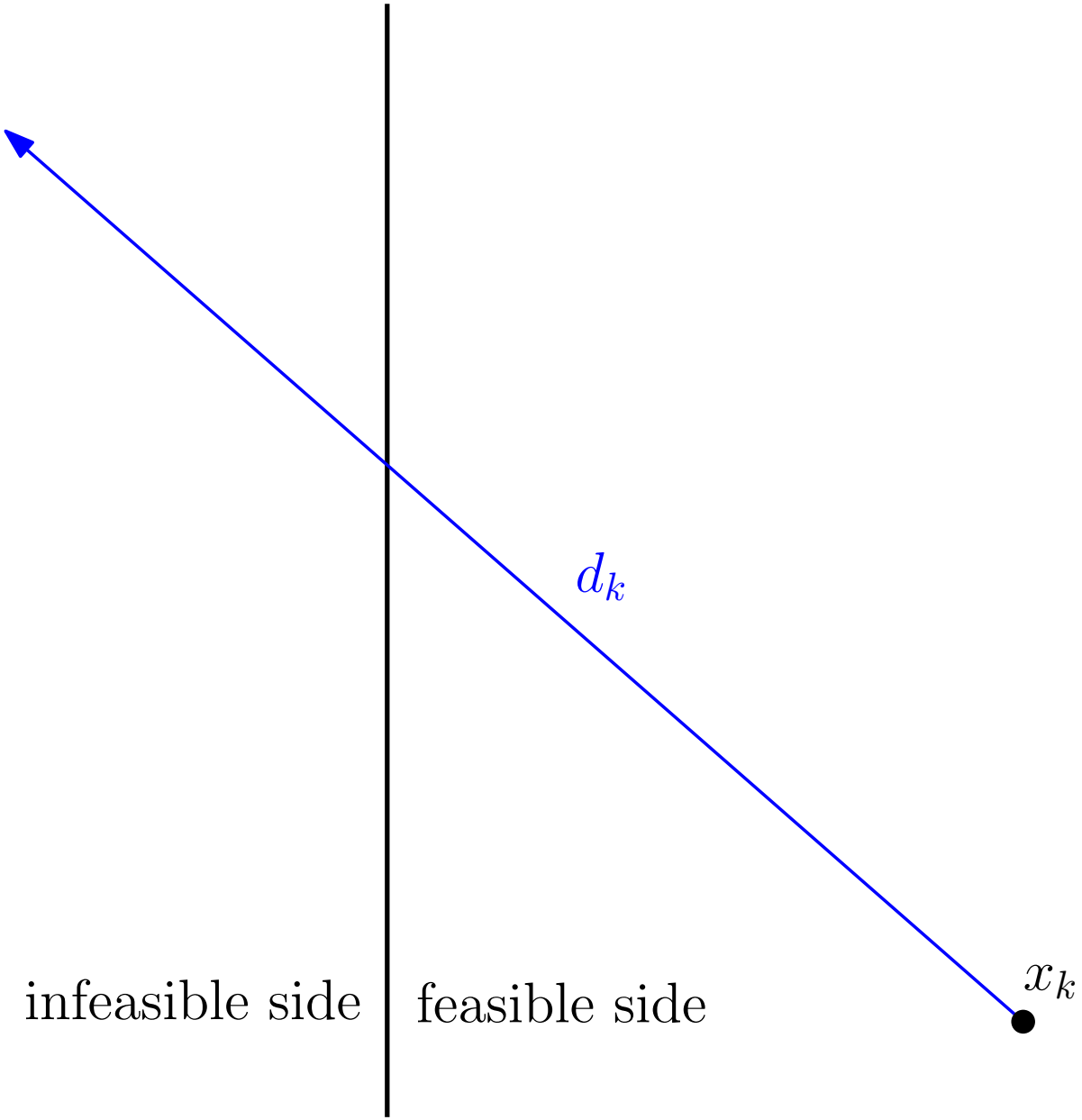
- Can prove that $\{\varphi(x_k)\}$ and $\{\beta(x_k)\}$ both converge to zero. (global convergence)
- Computing a direction $\bar{d}_k$ satisfying $g_k^T \bar{d}_k \leq g_k^T d_k^R$ and $m_k(\bar{d}_k) \leq m_k(0)$.
  - The Newton step $H_k d = -g_k$.
  - Any iteration computed via CG.
  - Iterations of coordinate descent (asymptotically).
- Algorithm LINESEARCH_φ is not a standard backtracking linesearch:

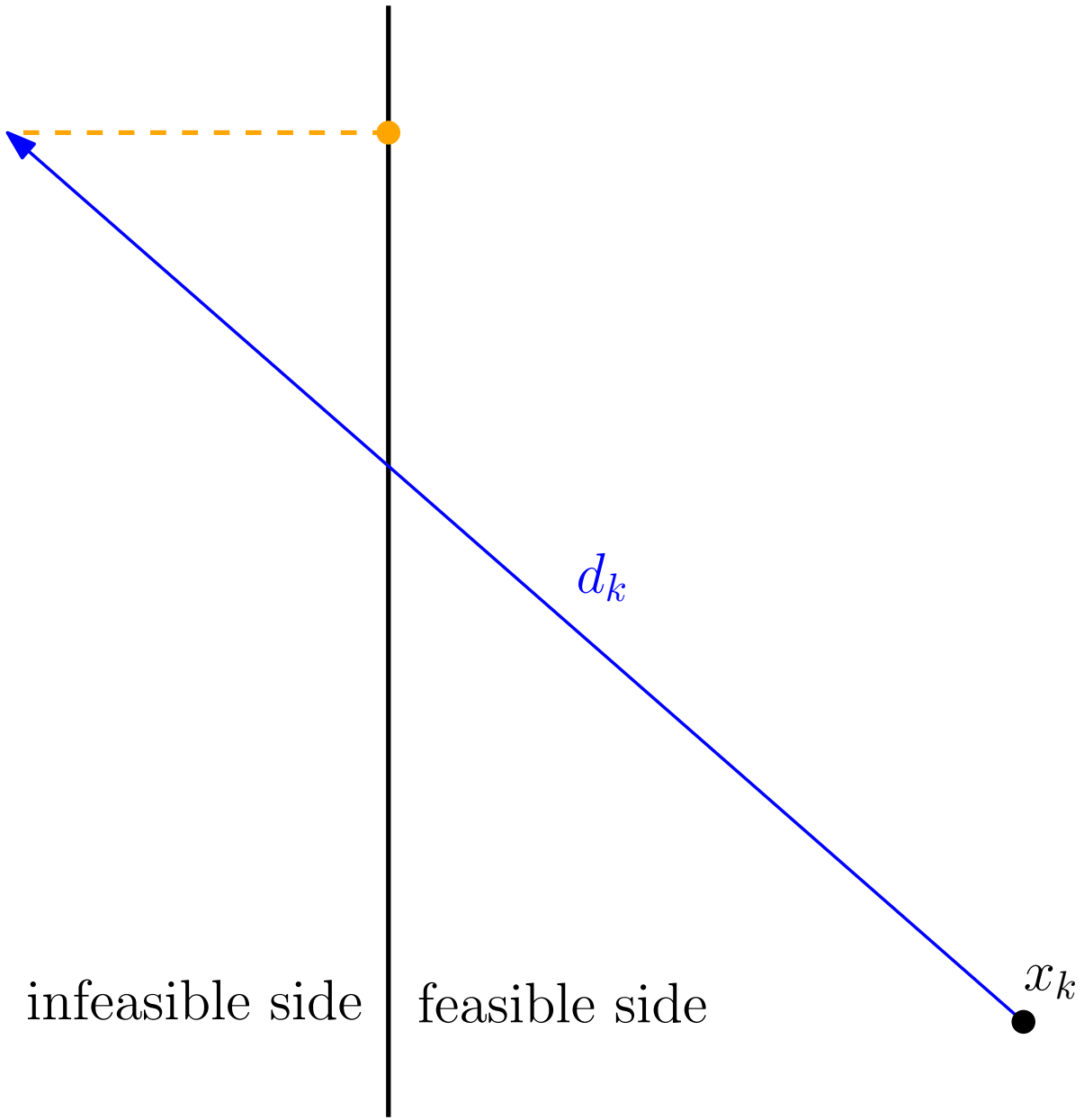
$$\text{(new variable becomes active) and } F(x_{k+1}) \leq F(x_k)$$
$$\text{or}$$
$$\text{(no new variables become active) and } F(x_{k+1}) \leq F(x_k) - \eta\alpha_k \nabla F(x_k)^T d_k$$

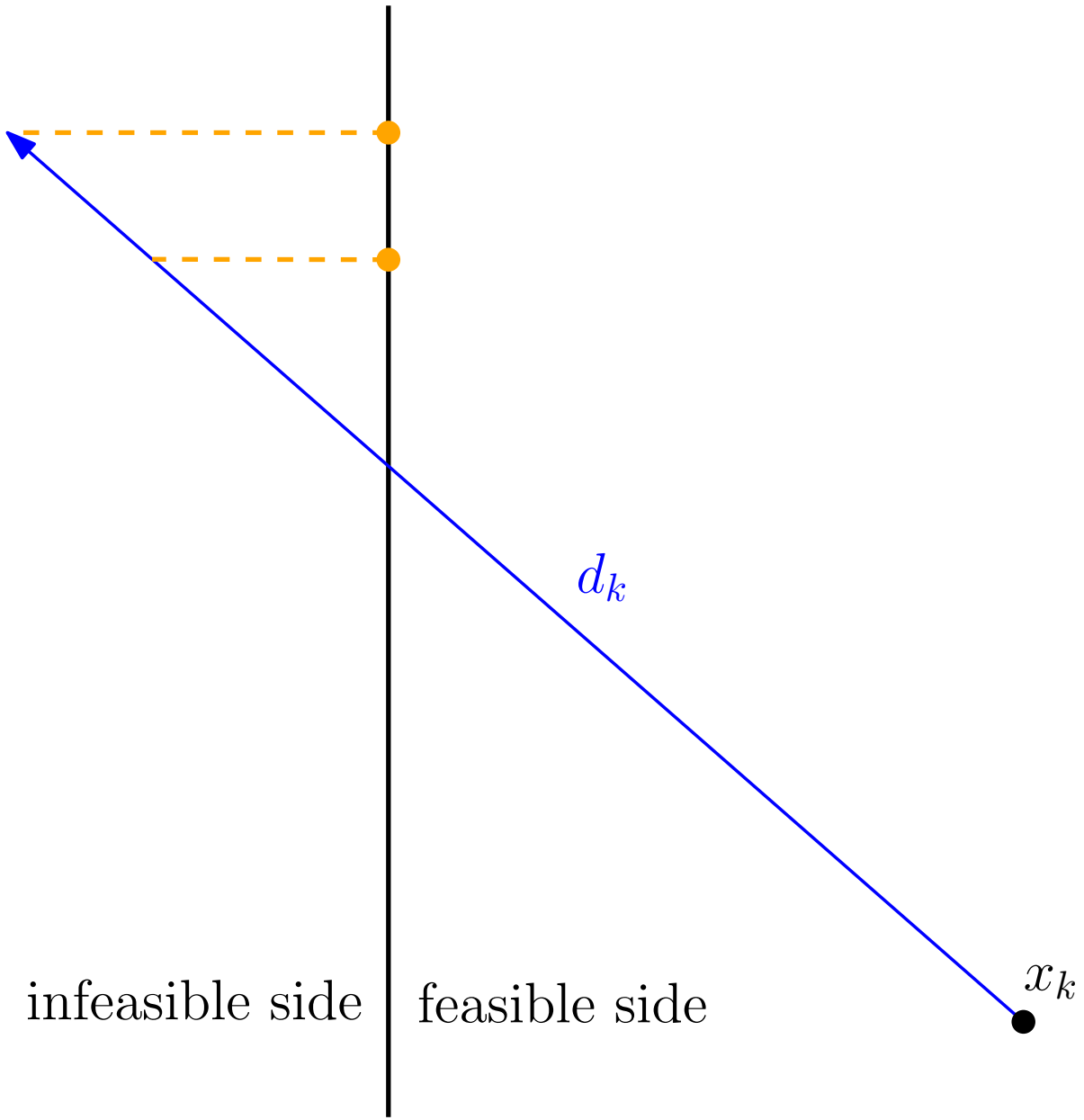
- Algorithm LINESEARCH_φ may only fix additional variables. (SRP)
- Algorithm LINESEARCH_β is a standard backtracking-Armijo linesearch:

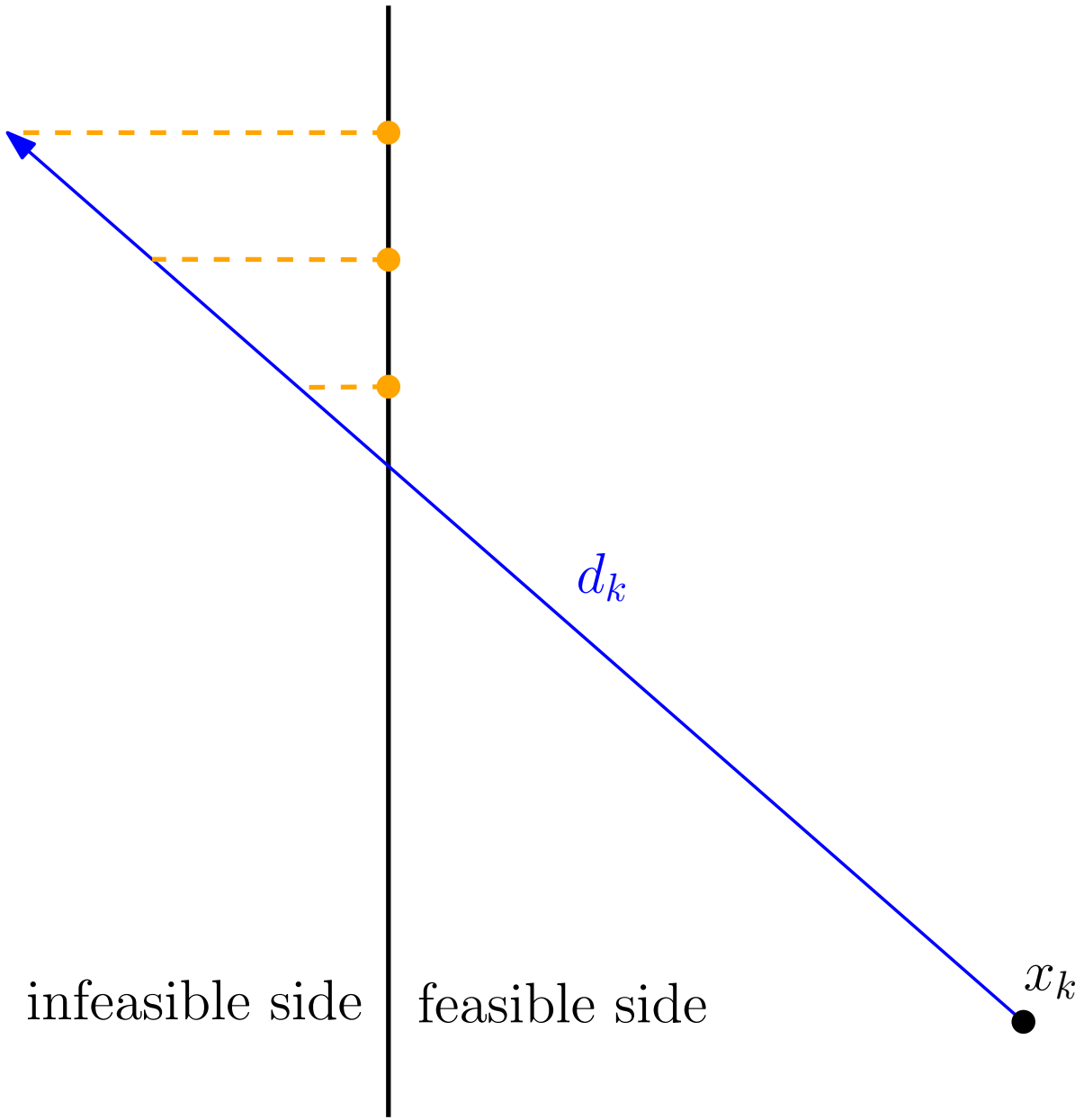
$$F(x_{k+1}) \leq F(x_k) - \eta\alpha_k \|d_k\|^2$$

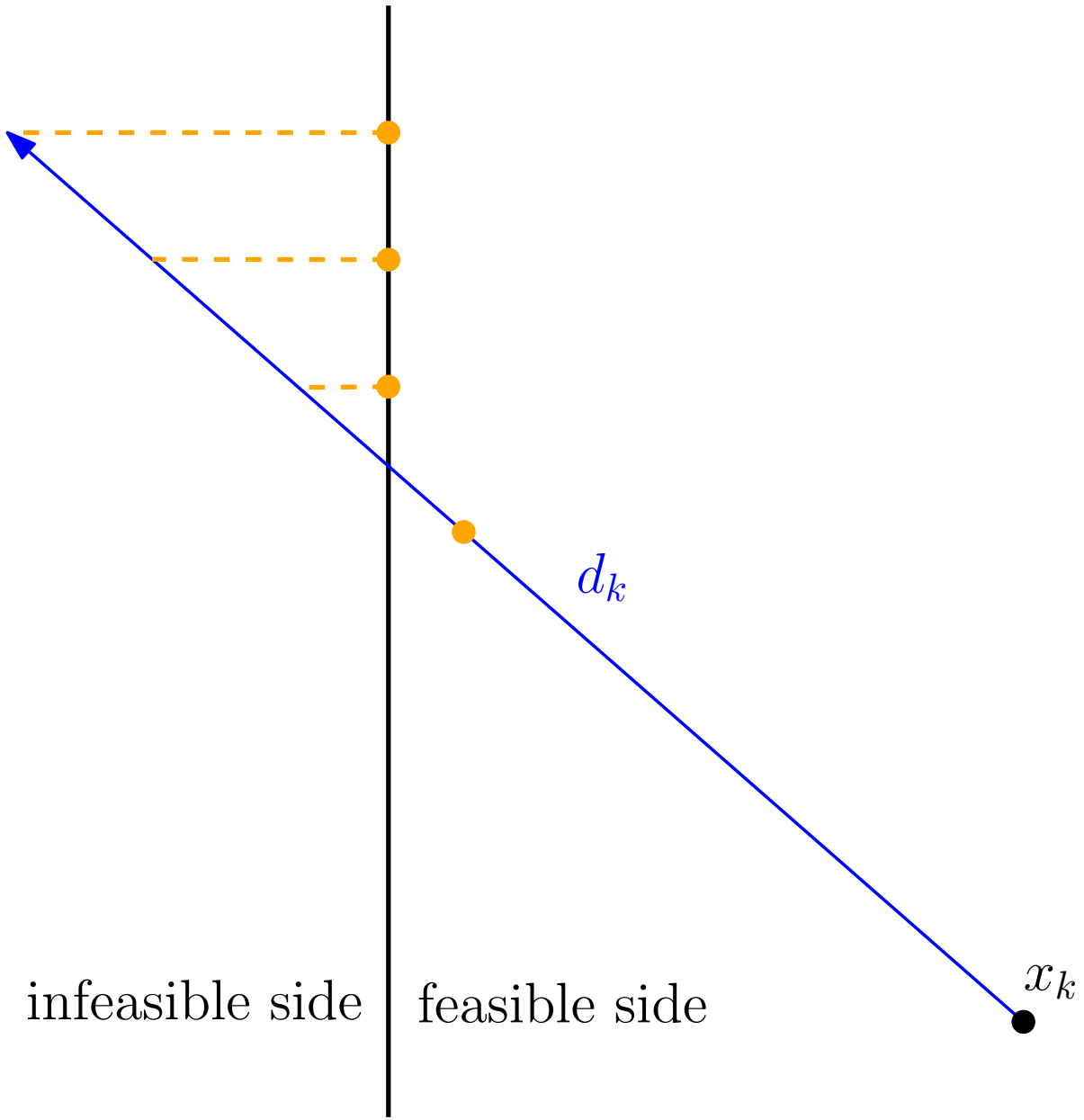
- Algorithm LINESEARCH_β only frees up some fixed variables. (SRP)
- Choice of $\mathcal{I}_k$ allows us to control the size of the subproblems.
- Overall, very flexible subproblem formulation and choice of subproblem solver.

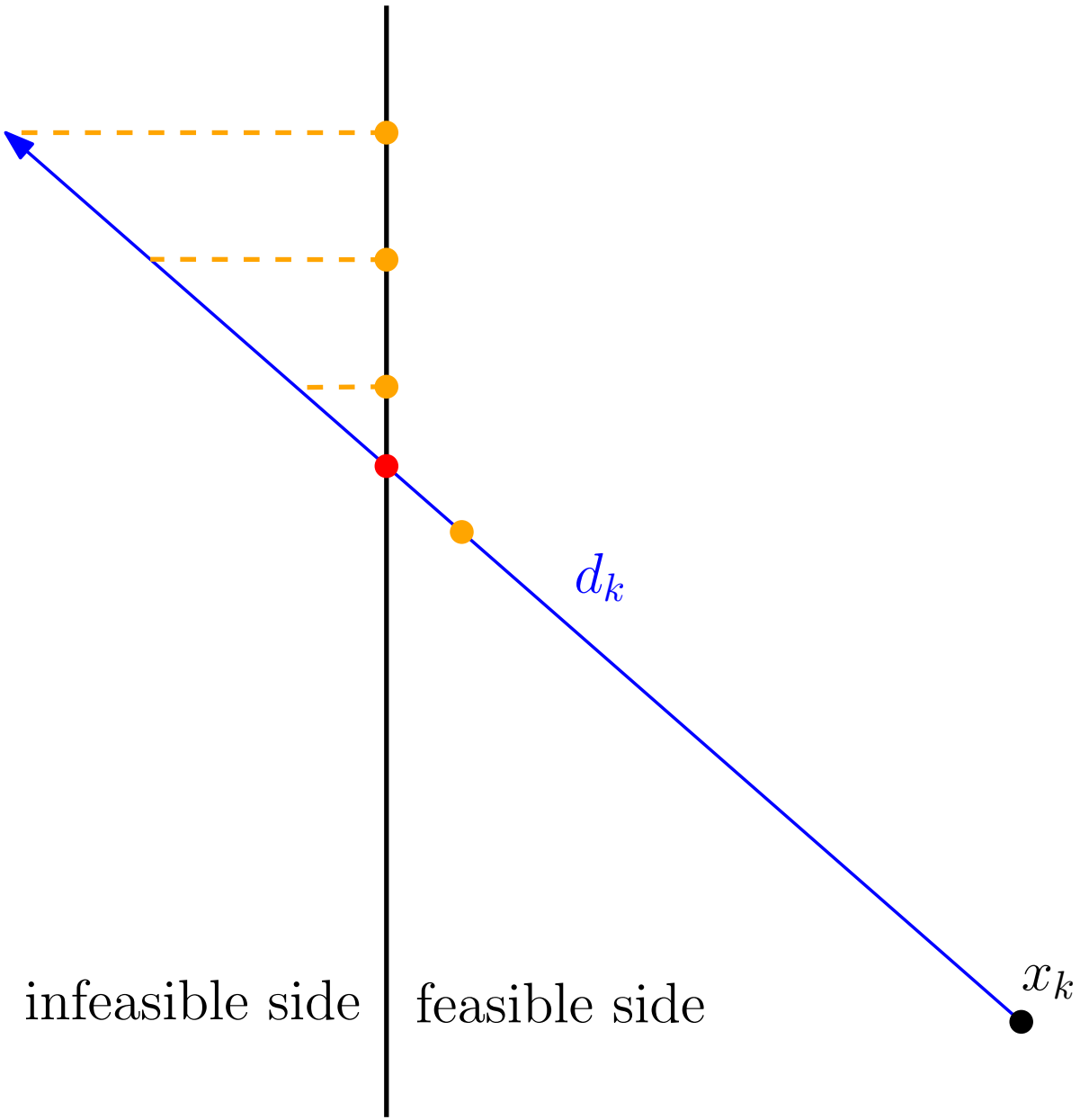

d_k
infeasible side
feasible side
x_k


d_k
x_k
infeasible side
feasible side


d_k
x_k
infeasible side
feasible side


d_k
x_k
infeasible side
feasible side


d_k
x_k
infeasible side
feasible side


d_k
infeasible side
feasible side
x_k

Definition (Zero, positive, and negative variables at x)

$$\mathcal{I}^0(x) := \{i : [x]_i = 0\}, \quad \mathcal{I}^+(x) := \{i : [x]_i > 0\}, \quad \text{and} \quad \mathcal{I}^-(x) := \{i : [x]_i < 0\}.$$

Definition (Zero, positive, and negative variables at x)

$$\mathcal{I}^0(x) := \{i : [x]_i = 0\}, \quad \mathcal{I}^+(x) := \{i : [x]_i > 0\}, \quad \text{and} \quad \mathcal{I}^-(x) := \{i : [x]_i < 0\}.$$

Definition (Reduced free-gradient and fixed-gradient)

$$[\phi(x)]_i = \begin{cases} 0 & \text{if } i \in \mathcal{I}^0 \\ \min\{[\nabla f(x)]_i + \lambda, \max\{[x]_i, [\nabla f(x)]_i - \lambda\}\} & \text{if } i \in \mathcal{I}^+ \text{ and } [\nabla f(x)]_i + \lambda > 0 \\ \max\{[\nabla f(x)]_i - \lambda, \min\{[x]_i, [\nabla f(x)]_i + \lambda\}\} & \text{if } i \in \mathcal{I}^- \text{ and } [\nabla f(x)]_i - \lambda < 0 \\ [\nabla f(x) + \lambda \cdot \operatorname{sgn}(x)]_i & \text{otherwise} \end{cases}$$

$$[\beta(x)]_i = \begin{cases} [\nabla f(x)]_i + \lambda & \text{if } i \in \mathcal{I}^0(x) \text{ and } [\nabla f(x)]_i + \lambda < 0 \\ [\nabla f(x)]_i - \lambda & \text{if } i \in \mathcal{I}^0(x) \text{ and } [\nabla f(x)]_i - \lambda > 0 \\ 0 & \text{otherwise} \end{cases}$$

Definition (Zero, positive, and negative variables at x)

$$\mathcal{I}^0(x) := \{i : [x]_i = 0\}, \quad \mathcal{I}^+(x) := \{i : [x]_i > 0\}, \quad \text{and} \quad \mathcal{I}^-(x) := \{i : [x]_i < 0\}.$$

Definition (Reduced free-gradient and fixed-gradient)

$$[\phi(x)]_i = \begin{cases} 0 & \text{if } i \in \mathcal{I}^0 \\ \min\{[\nabla f(x)]_i + \lambda, \max\{[x]_i, [\nabla f(x)]_i - \lambda\}\} & \text{if } i \in \mathcal{I}^+ \text{ and } [\nabla f(x)]_i + \lambda > 0 \\ \max\{[\nabla f(x)]_i - \lambda, \min\{[x]_i, [\nabla f(x)]_i + \lambda\}\} & \text{if } i \in \mathcal{I}^- \text{ and } [\nabla f(x)]_i - \lambda < 0 \\ [\nabla f(x) + \lambda \cdot \operatorname{sgn}(x)]_i & \text{otherwise} \end{cases}$$

$$[\beta(x)]_i = \begin{cases} [\nabla f(x)]_i + \lambda & \text{if } i \in \mathcal{I}^0(x) \text{ and } [\nabla f(x)]_i + \lambda < 0 \\ [\nabla f(x)]_i - \lambda & \text{if } i \in \mathcal{I}^0(x) \text{ and } [\nabla f(x)]_i - \lambda > 0 \\ 0 & \text{otherwise} \end{cases}$$

Lemma (relationship to ISTA)

The ISTA step $s_k^{ISTA} := \operatorname{shrink}\big[x_k - \nabla f(x_k)\big] - x_k$ *satisfies*

$$s_k^{ISTA} \equiv -\big(\beta(x_k) + \varphi(x_k)\big)$$

Outline

UCI Machine Learning Repository

Five methods:

1. FaRSA-CG (FAst Reduced Space Algorithm)
2. FaRSA-CD
3. FaRSA-Cross
4. OBA (Keskar, Nocedal, Öztoprak, and Wächter)
5. LIBLINEAR (Fan, Chang, Hsieh, Wang, and Lin)

Some choices:

- $\lambda = 1/N$, where N is the number of data.
- Stopping criteria:
  - Absolute: $\max\{\|\beta(x_k)\|, \|\phi(x_k)\|\} \leq 10^{-6}$
  - Relative: $\max\{\|\beta(x_k)\|, \|\phi(x_k)\|\} \leq 10^{-6} \max\{1, \|\beta(x_0)\|, \|\phi(x_0)\|\}$

UCI Machine Learning Repository

Test Set Summary

Dataset	N	n	λ	Scaled	Unscaled
a9a	32561	123	0.000031	✓	
Breast Cancer	681	10	0.1464	✓	
Diabetes	681	10	0.1464	✓	✓
gene AD	71	17375	0.014085	✓	✓
gene AD2	123	17375	0.008130	✓	✓
german numer	1000	24	0.001	✓	✓
ijcnn1	49990	22	0.0002	✓	
leu	38	7129	0.026316	✓	
mnist	30000	780	0.000033	✓	✓
pathway AD	71	278	0.014085	✓	
real sim	72309	20958	0.000014	✓	
w8a	49749	300	0.00002	✓	
gisette	6000	5000	0.000167	✓	

FaRSA: relative stopping criteria

		FaRSA-CG			FaRSA-CD			FaRSA-Cross		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	37	3.841	0.32	max			42	5.351	0.32
breast cancer	✓	21	0.035	0.09	41	0.196	0.09	20	0.237	0.09
diabetes	✓	14	0.208	0.54	21	0.183	0.54	14	0.158	0.54
gene AD	✓	51	3.076	0.37	178	13.10	0.37	119	10.05	0.37
gene AD2	✓	84	4.625	0.30	197	15.12	0.30	67	10.41	0.30
german numer	✓	20	0.170	0.58	42	0.339	0.58	20	0.650	0.58
ijcnn1	✓	18	0.697	0.21	46	2.238	0.21	18	0.611	0.21
leu	✓	83	3.786	0.17	83	2.908	0.17	45	2.426	0.17
mnist	✓	31	59.20	0.33	309	88.64	0.33	37	60.37	0.33
pathway AD	✓	154	0.737	0.27	max			159	0.948	0.27
real sim	✓	23	25.71	0.14	24	20.88	0.14	23	25.87	0.14
w8a	✓	56	8.906	0.12	73	6.982	0.12	40	5.363	0.12
gisette	✓	244	432.4	0.00	39	7.754	0.00	198	284.6	0.00
diabetes		20	0.066	0.60	162	0.376	0.60	57	0.193	0.60
gene AD		87	5.589	0.00	233	8.459	0.00	258	10.05	0.00
gene AD2		141	13.12	0.02	max			max		
german numer		40	0.107	0.48	301	0.764	0.48	61	0.162	0.48
mnist		101	578.8	0.32	455	169.1	0.32	359	462.2	0.32

FaRSA: relative stopping criteria

		FaRSA-CG			FaRSA-CD			FaRSA-Cross		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	37	3.841	0.32	max			42	5.351	0.32
breast cancer	✓	21	0.035	0.09	41	0.196	0.09	20	0.237	0.09
diabetes	✓	14	0.208	0.54	21	0.183	0.54	14	0.158	0.54
gene AD	✓	51	3.076	0.37	178	13.10	0.37	119	10.05	0.37
gene AD2	✓	84	4.625	0.30	197	15.12	0.30	67	10.41	0.30
german numer	✓	20	0.170	0.58	42	0.339	0.58	20	0.650	0.58
ijcnn1	✓	18	0.697	0.21	46	2.238	0.21	18	0.611	0.21
leu	✓	83	3.786	0.17	83	2.908	0.17	45	2.426	0.17
mnist	✓	31	59.20	0.33	309	88.64	0.33	37	60.37	0.33
pathway AD	✓	154	0.737	0.27	max			159	0.948	0.27
real sim	✓	23	25.71	0.14	24	20.88	0.14	23	25.87	0.14
w8a	✓	56	8.906	0.12	73	6.982	0.12	40	5.363	0.12
gisette	✓	244	432.4	0.00	39	7.754	0.00	198	284.6	0.00
diabetes		20	0.066	0.60	162	0.376	0.60	57	0.193	0.60
gene AD		87	5.589	0.00	233	8.459	0.00	258	10.05	0.00
gene AD2		141	13.12	0.02	max			max		
german numer		40	0.107	0.48	301	0.764	0.48	61	0.162	0.48
mnist		101	578.8	0.32	455	169.1	0.32	359	462.2	0.32

Comments:

- CG is better on 80% (54%) of unscaled (scaled) problems.
- No single strategy seems to be the best.
- A combination of subproblem solvers (in parallel?) is probably best.

FaRSA versus OBA: absolute stopping criteria

		FaRSA-CG			OBA		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	40	4.490008	0.32428	25	7.14697	0.32428
breast cancer	✓	21	0.027819	0.09812	9	0.33276	0.09812
diabetes	✓	14	0.020580	0.54587	6	0.183343	0.54587
gene AD	✓	52	1.680945	0.37386	9	1.258485	0.37386
gene AD2	✓	85	3.864362	0.30305	10	1.767881	0.30305
german numer	✓	20	0.040332	0.58488	8	0.170253	0.58488
ijcnn1	✓	18	0.183396	0.21391	8	0.7141	0.21391
leu	✓	83	1.527173	0.17995	15	2.218227	0.17995
mnist	✓	33	68.024364	0.33308	43	68.544698	0.33308
pathway AD	✓	156	1.239065	0.27328	88	8.081732	0.27328
real sim	✓	23	27.844184	0.14125	10	4.932584	0.14125
w8a	✓	57	9.118364	0.12206	14	4.524576	0.12206
diabetes		21	0.044525	0.60913	9	0.208821	0.60913
gene AD		473	35.090564	0.00080	fail(ascent)		
gene AD2		max			fail(ascent)		
german numer		40	0.10733	0.48005	17	0.376286	0.48005

FaRSA versus OBA: absolute stopping criteria

		FaRSA-CG			OBA		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	40	4.490008	0.32428	25	7.14697	0.32428
breast cancer	✓	21	0.027819	0.09812	9	0.33276	0.09812
diabetes	✓	14	0.020580	0.54587	6	0.183343	0.54587
gene AD	✓	52	1.680945	0.37386	9	1.258485	0.37386
gene AD2	✓	85	3.864362	0.30305	10	1.767881	0.30305
german numer	✓	20	0.040332	0.58488	8	0.170253	0.58488
ijcnn1	✓	18	0.183396	0.21391	8	0.7141	0.21391
leu	✓	83	1.527173	0.17995	15	2.218227	0.17995
mnist	✓	33	68.024364	0.33308	43	68.544698	0.33308
pathway AD	✓	156	1.239065	0.27328	88	8.081732	0.27328
real sim	✓	23	27.844184	0.14125	10	4.932584	0.14125
w8a	✓	57	9.118364	0.12206	14	4.524576	0.12206
diabetes		21	0.044525	0.60913	9	0.208821	0.60913
gene AD		473	35.090564	0.00080	fail(ascent)		
gene AD2		max			fail(ascent)		
german numer		40	0.10733	0.48005	17	0.376286	0.48005

Comments:

- FaRSA (OBA) is better on 73% (33%) of the problems that could be solved.
- FaRSA usually no more than 2-times worse (real-sim is about 5-times worse).
- FaRSA about 10-times better on breast-cancer and diabetes (scaled).

FaRSA versus LIBLINEAR: relative stopping condition

		FaRSA-CG			LIBLINEAR		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	37	3.84119	0.32428	61	7.858267	0.32428
breast cancer	✓	21	0.03523	0.09812	26	0.006978	0.09812
diabetes	✓	14	0.20803	0.54587	16	0.002716	0.54587
gene AD	✓	51	3.07643	0.37386	21	0.310244	0.37386
gene AD2	✓	84	4.62597	0.30305	18	0.384898	0.30305
german numer	✓	20	0.17050	0.58488	20	0.022464	0.58488
ijcnn1	✓	18	0.69700	0.21391	28	0.292693	0.21391
leu	✓	83	3.78696	0.17995	26	0.121428	0.17995
mnist	✓	31	59.20399	0.33308	22	8.802558	0.3331
pathway AD	✓	154	0.73740	0.27328	98	0.442351	0.27338
real sim	✓	23	25.71329	0.14125	18	2.5241	0.14125
w8a	✓	56	8.90658	0.12206	34	1.398444	0.12206
gisette	✓	244	432.49533	0.00013	25	5.0784	0.00014
diabetes		20	0.06607	0.60913	42	0.016202	0.60913
gene AD		87	5.58989	0.00080	21	0.245292	0.001
gene AD2		141	13.12783	0.02313	303	12.210618	0.02313
german numer		40	0.10733	0.48005	45	0.040327	0.48005
mnist		101	578.83868	0.32253	35	29.237833	0.32253

FaRSA versus LIBLINEAR: relative stopping condition

		FaRSA-CG			LIBLINEAR		
Dataset	Scaled	Iters.	Secs.	Objective	Iters.	Secs.	Objective
a9a	✓	37	3.84119	0.32428	61	7.858267	0.32428
breast cancer	✓	21	0.03523	0.09812	26	0.006978	0.09812
diabetes	✓	14	0.20803	0.54587	16	0.002716	0.54587
gene AD	✓	51	3.07643	0.37386	21	0.310244	0.37386
gene AD2	✓	84	4.62597	0.30305	18	0.384898	0.30305
german numer	✓	20	0.17050	0.58488	20	0.022464	0.58488
ijcnn1	✓	18	0.69700	0.21391	28	0.292693	0.21391
leu	✓	83	3.78696	0.17995	26	0.121428	0.17995
mnist	✓	31	59.20399	0.33308	22	8.802558	0.3331
pathway AD	✓	154	0.73740	0.27328	98	0.442351	0.27338
real sim	✓	23	25.71329	0.14125	18	2.5241	0.14125
w8a	✓	56	8.90658	0.12206	34	1.398444	0.12206
gisette	✓	244	432.49533	0.00013	25	5.0784	0.00014
diabetes		20	0.06607	0.60913	42	0.016202	0.60913
gene AD		87	5.58989	0.00080	21	0.245292	0.001
gene AD2		141	13.12783	0.02313	303	12.210618	0.02313
german numer		40	0.10733	0.48005	45	0.040327	0.48005
mnist		101	578.83868	0.32253	35	29.237833	0.32253

Comments:

- LIBLINEAR is better on 100% of the problems.
- factor worse: 2-times (4), 5-times (5), 10-times (7), 100-times (2).
- Why? Partly explained by MATLAB versus C++ implementation of CD.

Outline

Summary:

- Our method uses ASM and SRP computations. Shares ideas with other methods:
  - orthant based ideas (to some extent)
  - active-set strategy
  - ISTA step (in reduced space of active variables)
  - every step guarantees "progress" toward a solution
- We have a basic global convergence result under strong assumptions.
- Subproblem termination conditions allow for Newton-CG and CD solvers.
- Numerical results are preliminary, but promising.

Future Work:

- Extensive testing on large-scale problems.
- Can we use two-metric ideas to improve active-gradient step? We think so!
- Global convergence when two-metric ideas are used.
- Practical considerations such as limiting the subproblem size.
- Efficient CD implementation! Use ideas from LIBLINEAR?
- Run subproblem solvers in parallel; use first to complete.
- Rate-of-convergence.

Thank You