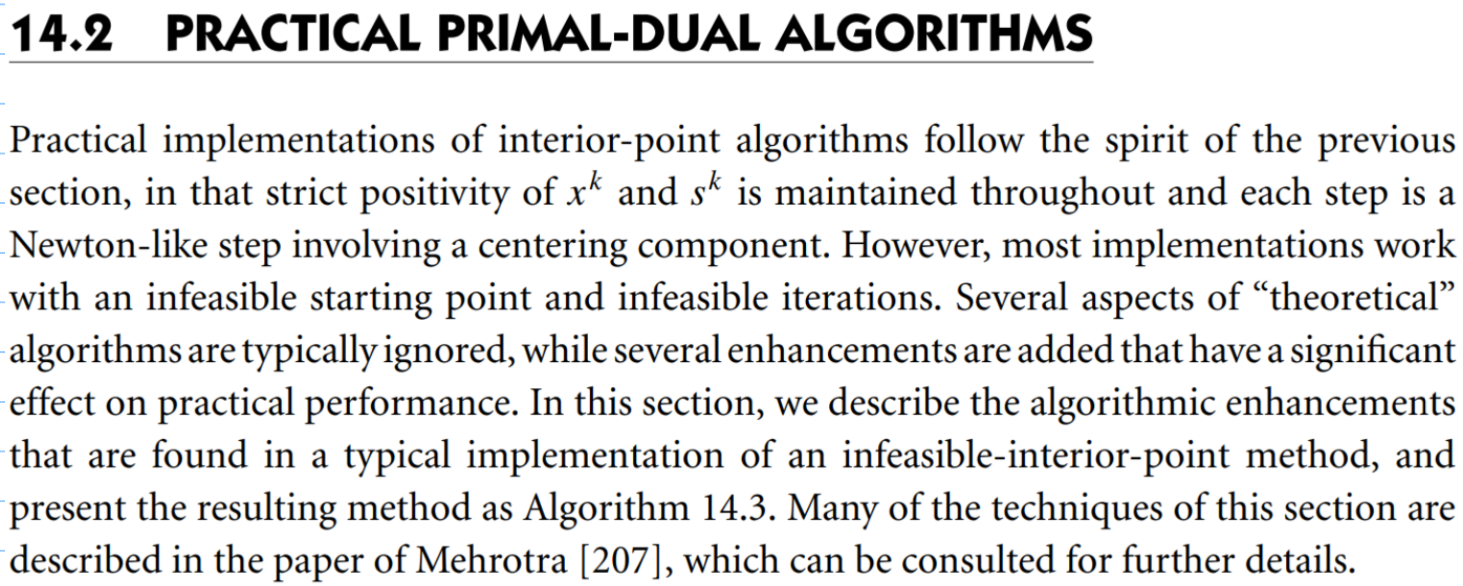


14.2 PRACTICAL PRIMAL-DUAL ALGORITHMS

Practical implementations of interior-point algorithms follow the spirit of the previous section, in that strict positivity of x^k and s^k is maintained throughout and each step is a Newton-like step involving a centering component. However, most implementations work with an infeasible starting point and infeasible iterations. Several aspects of "theoretical" algorithms are typically ignored, while several enhancements are added that have a significant effect on practical performance. In this section, we describe the algorithmic enhancements that are found in a typical implementation of an infeasible-interior-point method, and present the resulting method as Algorithm 14.3. Many of the techniques of this section are described in the paper of Mehrotra [207], which can be consulted for further details.

Notice that the proof assumes that the initial point (x^0, λ^0, s^0), and hence all subsequent iterates (x^k, λ^k, s^k), are feasible. This also means

$$r_c^k = 0, \quad r_b^k = 0 \quad \text{for all } k.$$

Practical solvers do not assume this.

Outline of a single iteration of a practical algorithm:

Idea #1: Adaptive choice of centering parameter σ

Given a point (x, λ, s), $x, s > 0$ (possibly infeasible), the pure Newton (aka "affine-scaling") direction is found by solving:

$$\text{(AS)} \quad \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta \lambda^{\text{aff}} \\ \Delta s^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix} \quad \begin{aligned} &\equiv A^T\lambda + s - c \\ &\equiv Ax - b \end{aligned}$$

This is our (14.10) with $\sigma = 0$.

To the very least, we should damp the Newton step to maintain positivity.

$$\alpha_{\text{aff}}^{\text{pri}} = \operatorname{argmax}\{\alpha \in [0,1] : x + \alpha \Delta x^{\text{aff}} \geq 0\} = \min\Big(1, \min_{i : \Delta x_i^{\text{aff}} < 0} -x_i / \Delta x_i^{\text{aff}}\Big)$$

$$\alpha_{\text{aff}}^{\text{dual}} = \operatorname{argmax}\{\alpha \in [0,1] : s + \alpha \Delta s^{\text{aff}} \geq 0\} = \min\Big(1, \min_{i : \Delta s_i^{\text{aff}} < 0} -s_i / \Delta s_i^{\text{aff}}\Big)$$

Let $\mu_{\text{aff}} = (x + \alpha_{\text{aff}}^{\text{pri}} \Delta x^{\text{aff}})^T (s + \alpha_{\text{aff}}^{\text{dual}} \Delta s^{\text{aff}}) / n$

If $\mu_{\text{aff}} \ll \mu$, then the affine direction is good for reducing μ. Little centering is needed. Choose the centering parameter σ close to 0.

If $\mu_{\text{aff}} \approx \mu$, the current iterate is too close to the boundary. More centering is needed. Choose σ close to 1.

Based on this insight, Mehrotra suggests to choose σ adaptively by:

$$\sigma = (\mu_{aff}/\mu)^3.$$

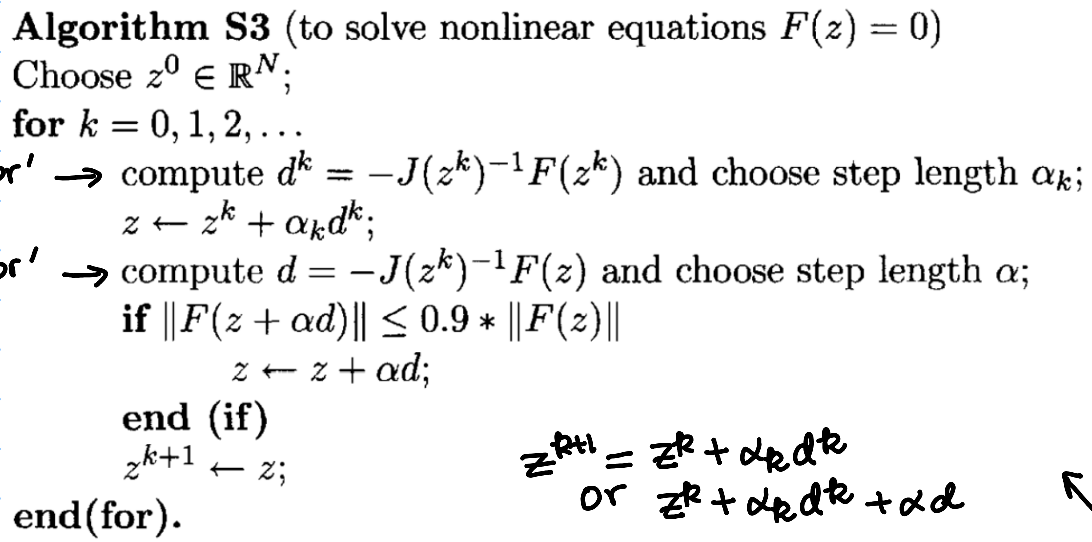
(This is a heuristical choice found to work well in practice. In fact, it is not clear to me if μ_{aff} is guaranteed to be less than μ all the time.)

We may proceed to solve (14.10) in order to determine the next iterate with this adaptive choice of centering parameter σ.

But this isn't what Mehrotra's predictor-corrector approach does.

Idea #2 : predictor - corrector

Precursor : an accelerated Newton's method based on "predictor - corrector".



```
Algorithm S3 (to solve nonlinear equations $F(z) = 0$)
Choose $z^0 \in \mathbb{R}^N$;
for $k = 0, 1, 2, \ldots$
`predictor' →   compute $d^k = -J(z^k)^{-1}F(z^k)$ and choose step length $\alpha_k$;
                $z \leftarrow z^k + \alpha_k d^k$;
`corrector' →   compute $d = -J(z^k)^{-1}F(z)$ and choose step length $\alpha$;
                if $\|F(z+\alpha d)\| \leq 0.9 * \|F(z)\|$
                    $z \leftarrow z + \alpha d$;
                end (if)
                $z^{k+1} \leftarrow z$;
end(for).
```

$z^{k+1} = z^k + \alpha_k d^k$

or $z^k + \alpha_k d^k + \alpha d$

$F : \mathbb{R}^N \to \mathbb{R}^N$

$J(\cdot)$ = Jacobian / Derivative = $DF(\cdot)$

Under suitable conditions :

Standard Newton satisfies quadratic convergence

$$\text{error}_{k+1} \lesssim C\, \text{error}_k^2$$

This version : $\text{error}_{k+1} \lesssim C\, \text{error}_k^3$.

Note : The two linear systems have the same coefficient matrix ($J(z^k)$).
Remember what it means computationally ?

Back to primal-dual interior point method :

Assume that we take a full Newton step, then

$$(x_i + \Delta x_i^{aff})(s_i + \Delta s_i^{aff}) = \underbrace{x_i s_i + x_i \Delta s_i^{aff} + s_i \Delta x_i^{aff}}_{=0 \text{ by the 3rd block row}} + \Delta x_i^{aff} \Delta s_i^{aff}$$

$$= \Delta x_i^{aff} \Delta s_i^{aff}$$

So the corrector step is to solve:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{cor} \\ \Delta \lambda^{cor} \\ \Delta s^{cor} \end{bmatrix} = -F(x + \Delta x^{aff}, \lambda + \Delta \lambda^{aff}, s + \Delta s^{aff}) = \begin{bmatrix} * \\ * \\ -\Delta X^{aff} \Delta S^{aff} e \end{bmatrix}$$

Mehrotra solves instead:

$$\Delta X^{aff} = \mathrm{diag}(\Delta x_1^{aff}, \dots, \Delta x_n^{aff})$$

$$\Delta S^{aff} = \mathrm{diag}(\Delta s_1^{aff}, \dots, \Delta s_n^{aff})$$

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{cor} \\ \Delta \lambda^{cor} \\ \Delta s^{cor} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Delta X^{aff} \Delta S^{aff} e \end{bmatrix}$$

← rhs cheap to compute, but things are getting ad hoc.

The combined step $(\Delta x^{aff}, \Delta \lambda^{aff}, \Delta s^{aff}) + (\Delta x^{cor}, \Delta \lambda^{cor}, \Delta s^{cor})$ usually does a better job of reducing the duality measure than does the affine-scaling step alone.

Putting Idea #1 and #2 together. (even more heuristical)

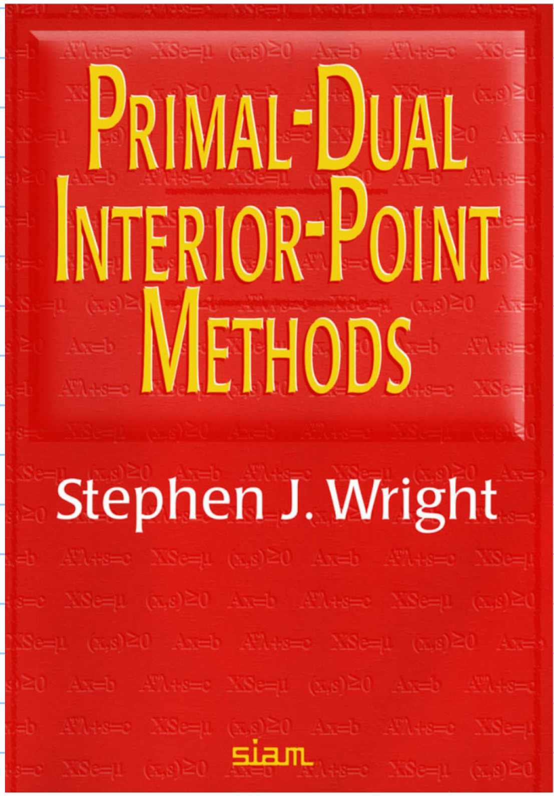
Mehrotra's centering-corrector step $(\Delta x^{cc}, \Delta \lambda^{cc}, \Delta s^{cc})$ is obtained by solving the following linear system:

$$\text{(CC)} \qquad \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{cc} \\ \Delta \lambda^{cc} \\ \Delta s^{cc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma \mu e - \Delta X^{aff} \Delta S^{aff} e \end{bmatrix}$$

Same coefficient matrix as in the affine-scaling step

the adaptively chosen centering parameter from #1, the computation of which requires first solving the pure-Newton / affine-scaling step

Given the practical success of this approach, a lot of follow-up work was done to understand / streamline the method.



```
Algorithm MPC
Given $(x^0, \lambda^0, s^0)$ with $(x^0, s^0) > 0$;
for $k = 0, 1, 2, \ldots$
    set $(x, \lambda, s) = (x^k, \lambda^k, s^k)$ and solve (10.1) (AS) for $(\Delta x^{\text{aff}}, \Delta\lambda^{\text{aff}}, \Delta s^{\text{aff}})$;
    calculate
        $\alpha^{\text{pri}}_{\text{aff}} = \arg\max\{\alpha \in [0,1] \,|\, x^k + \alpha\Delta x^{\text{aff}} \geq 0\}$,   (10.8a)
        $\alpha^{\text{dual}}_{\text{aff}} = \arg\max\{\alpha \in [0,1] \,|\, s^k + \alpha\Delta s^{\text{aff}} \geq 0\}$,   (10.8b)
        $\mu_{\text{aff}} = (x^k + \alpha^{\text{pri}}_{\text{aff}}\Delta x^{\text{aff}})^T(s^k + \alpha^{\text{dual}}_{\text{aff}}\Delta s^{\text{aff}})/n$;   (10.8c)
    set centering parameter to $\sigma = (\mu_{\text{aff}}/\mu)^3$;
    solve (10.7) (CC) for $(\Delta x^{\text{cc}}, \Delta\lambda^{\text{cc}}, \Delta s^{\text{cc}})$;
    compute search direction and step to boundary from
        $(\Delta x^k, \Delta\lambda^k, \Delta s^k) = (\Delta x^{\text{aff}}, \Delta\lambda^{\text{aff}}, \Delta s^{\text{aff}}) + (\Delta x^{\text{cc}}, \Delta\lambda^{\text{cc}}, \Delta s^{\text{cc}})$;   (10.9a)
        $\alpha^{\text{pri}}_{\max} = \arg\max\{\alpha \geq 0 \,|\, x^k + \alpha\Delta x^k \geq 0\}$;   (10.9b)
        $\alpha^{\text{dual}}_{\max} = \arg\max\{\alpha \geq 0 \,|\, s^k + \alpha\Delta s^k \geq 0\}$;   (10.9c)
    set $\alpha^{\text{pri}}_k = \min(0.99 * \alpha^{\text{pri}}_{\max}, 1)$ and $\alpha^{\text{dual}}_k = \min(0.99 * \alpha^{\text{dual}}_{\max}, 1)$;
    set
        $x^{k+1} = x^k + \alpha^{\text{pri}}_k \Delta x^k$,
        $(\lambda^{k+1}, s^{k+1}) = (\lambda^k, s^k) + \alpha^{\text{dual}}_k(\Delta\lambda^k, \Delta s^k)$;
end (for).
```

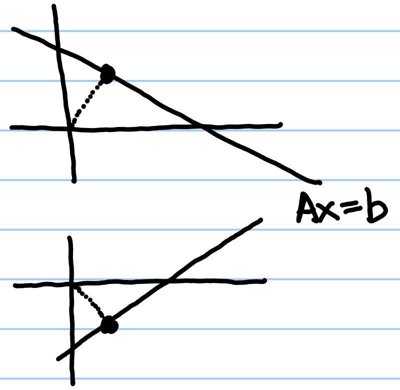
$$\boxed{\begin{aligned}&\operatorname{argmax}(\alpha \in [0,1] : v + \alpha\Delta v \geq 0)\\ &= \min\Big(1, \min_{i:\Delta v<0} -v_i/\Delta v_i\Big)\end{aligned}}$$

← The neighborhood $\mathcal{N}_{-\infty}(\gamma)$ in the long-step path following alg. is not used.

As defined above, Algorithm MPC is not quite the same as Mehrotra's original algorithm from [88]. However, it is similar to the variant that is implemented in codes such as LIPSOL, LOQO, and PCx (see Chapter 11).

Unlike the simplex method, it is less obvious how to detect unboundedness or infeasibility. See Ch 9 of the monograph.

Choice of starting point — see NBW, p410



Again pretty ad hoc, based on first choosing

$$\tilde{x} = \underset{Ax=b}{\operatorname{argmin}} \tfrac{1}{2} x^T x, \quad (\tilde{\lambda}, \tilde{s}) = \underset{A^T\lambda+s=c}{\operatorname{argmin}} \tfrac{1}{2} s^T s$$

$$= A^T(AA^T)^{-1}b \qquad \tilde{\lambda} = (AA^T)^{-1}Ac, \ \tilde{s} = c - A^T\tilde{\lambda} \quad ,$$

followed by adding a constant to them to ensure positivity and $x_i^0 s_i^0$ being not too dissimilar.

Solving the linear systems

Solving the linear systems

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -r_{xs} \end{bmatrix}$$

Since $x, s > 0$, X, S are non-singular. (And the whole coefficient matrix is non-singular, assuming A is full rank.

And this is another reason why the interior-point primal-dual method maintains positivity of x and s.)

In this case, we can eliminate Δs:

$$S \Delta x + X \Delta s = -r_{xs}$$
$$\Delta s = -X^{-1} r_{xs} - X^{-1} S \Delta x$$

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -r_c + X^{-1} r_{xs} \\ -r_b \end{bmatrix}, \quad D = S^{-1/2} X^{1/2}$$

(called augmented system)

(derived in class)

We may further eliminate Δx:

$$A D^2 A^T \Delta \lambda = -r_b - A X S^{-1} r_c + A S^{-1} r_{xs}, \quad \begin{aligned} \Delta s &= -r_c - A^T \Delta \lambda \\ \Delta x &= -S^{-1} r_{xs} - X S^{-1} \Delta s \end{aligned}$$

(called normal equations)

The normal equations are usually solved using Cholesky factorization

$M = PAD^2AP^T$, $M = LL^T$ P - a suitable permutation

$AD^2A\,d = y \iff M(Pd) = Py$

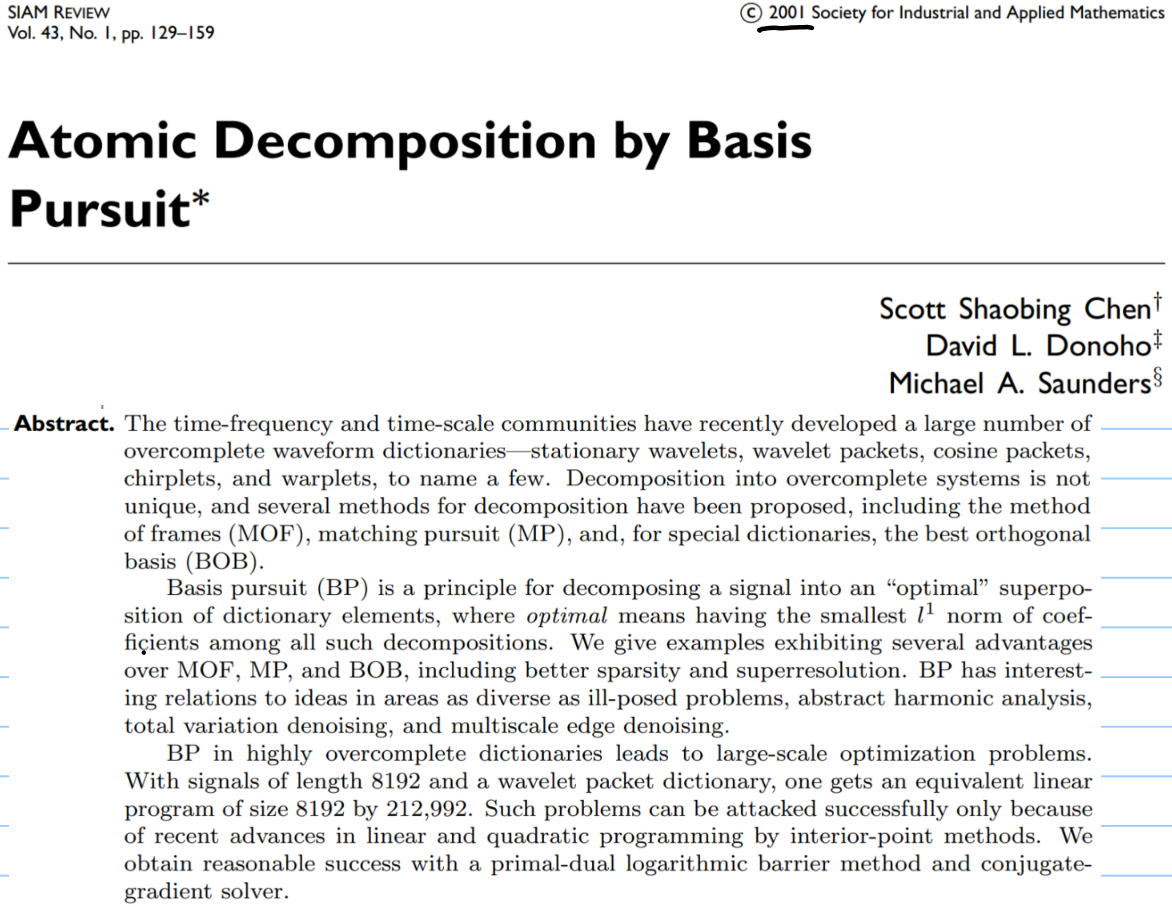
Solve $Lw = Py$ by forward substitution to find w
solve $L^Tz = w$ by backward substitution to find z
set $d = P^Tz$.

In practice, A is usually sparse and it is possible to find a good permutation P to reorder the equations and variables so that the L factor is also sparse. A lot of research was spent on figuring how to do this.

Also, a lot of techniques go into dealing with the possible ill-conditioning of AD^2A — especially when approaching the solution

$D^2_{ii} = x_i/s_i$ either goes to 0 or ∞ (assuming strict complimentarity, i.e. $x_i^*s_i^* = 0$, but not both $x_i^*, s_i^* = 0$)

The first time I saw a primal-dual point interior point method used in a serious application was in the following now-famous paper:



SIAM Review
Vol. 43, No. 1, pp. 129–159

Atomic Decomposition by Basis Pursuit*

Scott Shaobing Chen[†]
David L. Donoho[‡]
Michael A. Saunders[§]

Abstract. The time-frequency and time-scale communities have recently developed a large number of overcomplete waveform dictionaries—stationary wavelets, wavelet packets, cosine packets, chirplets, and warplets, to name a few. Decomposition into overcomplete systems is not unique, and several methods for decomposition have been proposed, including the method of frames (MOF), matching pursuit (MP), and, for special dictionaries, the best orthogonal basis (BOB).

Basis pursuit (BP) is a principle for decomposing a signal into an "optimal" superposition of dictionary elements, where *optimal* means having the smallest l^1 norm of coefficients among all such decompositions. We give examples exhibiting several advantages over MOF, MP, and BOB, including better sparsity and superresolution. BP has interesting relations to ideas in areas as diverse as ill-posed problems, abstract harmonic analysis, total variation denoising, and multiscale edge denoising.

BP in highly overcomplete dictionaries leads to large-scale optimization problems. With signals of length 8192 and a wavelet packet dictionary, one gets an equivalent linear program of size 8192 by 212,992. Such problems can be attacked successfully only because of recent advances in linear and quadratic programming by interior-point methods. We obtain reasonable success with a primal-dual logarithmic barrier method and conjugate-gradient solver.

In this application, the matrix A is not sparse. But it is 'sparse' in the sense that

$$x \mapsto Ax \ , \quad \lambda \mapsto A^T\lambda$$

can be computed in $O(\max(m,n))$ time, instead of $O(\max(m,n)^2)$ time for a general dense matrix. (Similar to the FFT.)

When m, n are large, the authors could not afford to compute the Cholesky factorization of AD^2A^T.

They used instead "matrix-free" conjugate gradient methods for solving the linear systems.

Compare

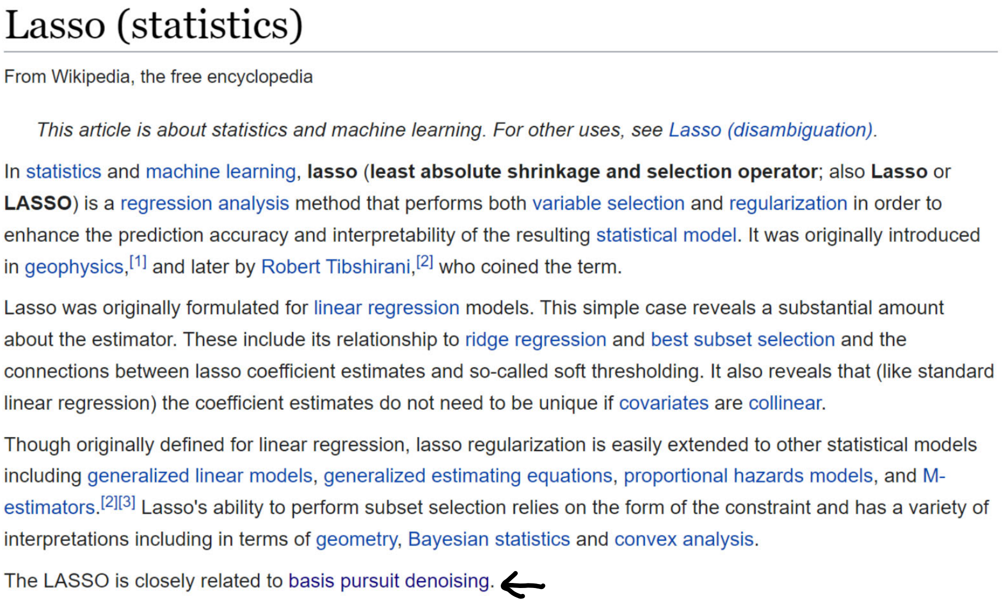
	$A \in \mathbb{R}^{m\times n}$ $m>n$		$A \in \mathbb{R}^{m\times n}$, $m<n$
Ordinary least square	$\min_x \|Ax-b\|_2$ $x_{LS} = (A^TA)^{-1}A^Tb$	Solution of $Ax=b$ with minimal 2-norm	$\min \|x\|_2$ s.t. $Ax=b$ $x^* = A^T(AA^T)^{-1}b$
min L^1 regression	$\min_x \|Ax-b\|_1$ a linear program	Solution of $Ax=b$ with minimal 1-norm (Basis Pursuit)	$\min \|x\|_1$ s.t. $Ax=b$ a linear program

Lasso $\approx$ OLS + BP

when $\underset{\substack{\uparrow\\ \text{sample size}}}{m} \ll \underset{\substack{\uparrow\\ \text{\# of features}}}{n}$

$$\min \|Ax-b\|_2 \quad \text{s.t.} \quad \underset{\substack{\uparrow\\ \text{promote sparsity}}}{\|x\|_1} \le t$$

A quadratic program



Lasso (statistics)

From Wikipedia, the free encyclopedia

This article is about statistics and machine learning. For other uses, see Lasso (disambiguation).

In statistics and machine learning, **lasso** (**least absolute shrinkage and selection operator**; also **Lasso** or **LASSO**) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model. It was originally introduced in geophysics,[1] and later by Robert Tibshirani,[2] who coined the term.

Lasso was originally formulated for linear regression models. This simple case reveals a substantial amount about the estimator. These include its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding. It also reveals that (like standard linear regression) the coefficient estimates do not need to be unique if covariates are collinear.

Though originally defined for linear regression, lasso regularization is easily extended to other statistical models including generalized linear models, generalized estimating equations, proportional hazards models, and M-estimators.[2][3] Lasso's ability to perform subset selection relies on the form of the constraint and has a variety of interpretations including in terms of geometry, Bayesian statistics and convex analysis.

The LASSO is closely related to basis pursuit denoising. ←