MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Parallel Quadratic Programming for Image Processing

Brand, M.; Chen, D.

TR2011-064 September 2011

Abstract

Many image processing and computer vision problems can be solved as quadratic programs in the non-negative cone. This paper develops a provably convergent multiplicative update that has a simple form and is amenable to fine-grained data parallelism. Classic algorithms for deblurring, matrix factorization, and tomography are recovered as special cases. This paper also demonstrates applications to super-resolution, labeling and segmentation.

IEEE International Conference on Image Processing (ICIP)

PARALLEL QUADRATIC PROGRAMMING FOR IMAGE PROCESSING

Matthew Brand

Mitsubishi Electric Research Labs
Cambridge, MA, USA

Donghui Chen[*]

Tufts University
Medford, MA, USA

ABSTRACT

Many image processing and computer vision problems can be solved as quadratic programs in the nonnegative cone. This paper develops a provably convergent multiplicative update that has a simple form and is amenable to fine-grained data parallelism. Classic algorithms for deblurring, matrix factorization, and tomography are recovered as special cases. This paper also demonstrates applications to super-resolution, labeling and segmentation.

***Index Terms*—** parallel quadratic programming, image super-resolution, image segmentation, Markov random field

1. INTRODUCTION

The non-negative quadratic program (NNQP)

$$\min_x \frac{1}{2} x^T Q x - x^T h \quad s.t. \quad x \geq 0, \tag{1}$$

—with vectors $x, h \in \mathscr{R}^n$ and symmetric positive semidefinite matrix $Q \in \mathscr{R}^{n \times n}$—is ubiquitous in image processing, where it is natural to reason about non-negative quantities such as light intensities under a squared error norm. For example, Eq. (1) subsumes the non-negative least squares problem (NNLS) encountered in deblurring and super-resolution,

$$\min_x ||Ax - b||_2^2 \quad s.t. \quad x \geq 0, \tag{2}$$

with $Q = A^T A$ and $h = A^T b$. Eq. (1) is also the dual of the broader class of convex quadratic programs

$$\min_y \frac{1}{2} y^T H y - y^T f \quad s.t. \quad Ay \geq b. \tag{3}$$

that arise in labeling/segmentation tasks and image operations that involve solving the Poisson equation (e.g., matting). Solutions of Eq. (1) yield solutions of Eq. (3).

A very large literature is devoted to Eqs. (1)–(3). State-of-the-art interior-point, active-set, and primal-dual algorithms offer linear-to-quadratic rates of convergence, but each iteration requires solution of a linear equation at least as large as $Qy = z$, which can take $O(n^3)$ time. This is not practical for image processing, where n may be as large as 10^7.

For that setting, we propose a multiplicative update algorithm derived from the Karush-Kuhn-Tucker (KKT, [1]) first-order optimality conditions of the Lagrangian of Eq. (1),

$$L(x, \lambda) = \frac{1}{2} x^\top Q x - x^\top h - x^\top \lambda$$

with $\lambda \in \mathscr{R}^n \geq 0$. These conditions are

$$\begin{aligned} x \circ \nabla_x L &= x \circ (Qx - h - \lambda) = 0 \\ \lambda \circ \nabla_\lambda L &= \lambda \circ x = 0 \end{aligned} \tag{4}$$

with $\circ$ denoting Hadamard (elementwise) product. The update arises from a variational treatment of these equations; in the limited space here, we will derive it directly from Eq. (4). Define the *matrix split* $Q = Q^+ - Q^-$ with $Q^+ = \max(Q, 0) + \operatorname{diag}(r)$ and $Q^- = \max(-Q, 0) + \operatorname{diag}(r)$ for some $r \in \mathscr{R}^n \geq 0$ (defined below). Similarly, $h = h^+ - h^-$ with $h^+ = \max(h, 0) + s$, $h^- = \max(-h, 0) + s$. Using the split, Eq. (4) can be written

$$x \circ \nabla_x L = x \circ ((Q^+ x + h^-) - (Q^- x + h^+ + \lambda)) = 0.$$

Assume for the moment that all $x_i > 0$, possibly infinitesimally. Then $\lambda_i = 0$ and, elementwise,

$$x_i (Q^+ x + h^-)_i = x_i (Q^+ x + h^-)_i.$$

Rearranging terms gives the fixpoint

$$x_i = x_i \left[\frac{h_i^+ + (Q^- x)_i}{h_i^- + (Q^+ x)_i} \right]. \tag{5}$$

which, by construction, is stationary at the optimum x^*.

We are interested in the behavior of this fixpoint as a multiplicative update for positive vectors $x \neq x^*$. Some properties are apparent from inspection:

- R.h.s. Eq. (5) can be calculated in parallel for all x_i by a simple matrix-vector multiply and a vectorized divide.
- The bracketed ratio is always non-negative (and positive if $s_i > 0$ or $r_i x_i > 0$). Thus, Eq. (1) preserves the feasibility of x, implicitly satisfying the second KKT condition.

[*] Chen performed this work while at MERL

- Like gradient descent, the fixpoint moves x_i in the direction opposite the partial derivative $\partial_{x_i} L(x, \lambda)$.

In fact, iteration of Eq. (5) from *any* $x > 0$ will solve of Eq. (1) and, by extension, Eq. (2)–(3):

Theorem 1.1 *For any $Q \succeq 0$ (positive semi-definite), there is an easily calculated vector r for which the fixpoint Eq.* (5) *converges monotonically and asymptotically from any positive $x > 0$ to the optimum of Eq.* (1).

A variational proof is given in a companion paper [2] where the update is used for high-speed optimal control of machinery. The role of r is to guarantee that the update gives a contraction even in the nullspace of semidefinite Q. There are many simple satisfactory choices of r, for example, $r_i = \max(Q_{ii}, \sum_j Q^-_{ij})$. For many strictly positive image processing problems it suffices to have $r = 0$. Note that convergence is asymptotic; if $x^*_i = 0$ we will only see $x_i \to 0$ (on an ideal infinite-precision computer). On real computers—which use floating point numbers—we see x_i rapidly converge to x^* regardless of its value.

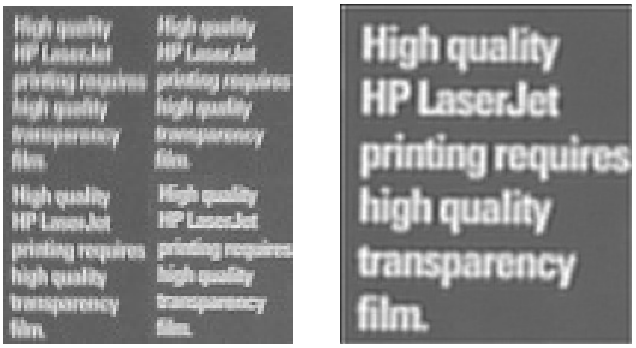
Furthermore, we prove below (Appendix A) that the asymptotic convergence rate is linear:

Theorem 1.2 *Asymptotically, the update of Eq.* (5) *improves the estimate $\hat{x}$ by a constant number of bits of precision.*

For the sparse Q matrices typically encountered in image processing, each update takes linear serial time or constant parallel time, and a linear number of updates are needed to achieve any fixed precision result. Thus, the multiplicative update combines the simplicity and calculational speed of gradient descent with the feasibility and convergence guarantees of modern quadratic programming algorithms. As such, it is particularly well suited for implementation on low-cost parallel processing arrays, such as the SIMD and GPU chips now going into mobile phones, especially since Q typically has low bandwidth in image processing tasks, so that the update requires very limited communication between processors. An exploratory GPU implementation solves NNQPs of 10^5 variables in less than a tenth of a second — three orders of magnitude faster than NNLS software on a multicore CPU.

2. RELATED WORK

Some well-known algorithms in the literature are recognizable as special cases of the update with simplifications for $Q \succ 0$ (positive definite), $Q \geq 0$ and/or $h \geq 0$. Dropping Q^- , h^-, and r yields the Richardson-Lucy deblurring algorithm and the image-space reconstruction algorithm [3]. If we formulate Non-negative Matrix Factorization as a pair of alternating least-squares optimizations, and use multiplicative update—absent Q^- and r—for each, we obtain the Lee-Seung algorithm [4]. Similar multiplicative fixpoint algorithms have been used successfully in tomography, astronomy (e.g., [5]), and estimation (e.g., [6]), however they rely on some combination of strictly nonnegative coefficients, positive definiteness, or favorable initialization for convergence, if convergence is provable at all. Our method is also related to matrix splitting algorithms for linear complementarity problems [7] and Uzawa methods for saddle-point problems [8]; unlike these methods, the update is given in closed form and can be calculated independently for each variable.



(a) 4 of 30 low-res images

(b) reconstructed high-res image

Fig. 1. Super-resolution example: a) 4 of 30 low-res images; b) the reconstruction high-res image obtained by solving SR model (6) with the multiplicative update.

3. IMAGE PROCESSING APPLICATIONS

3.1. Image Super-Resolution

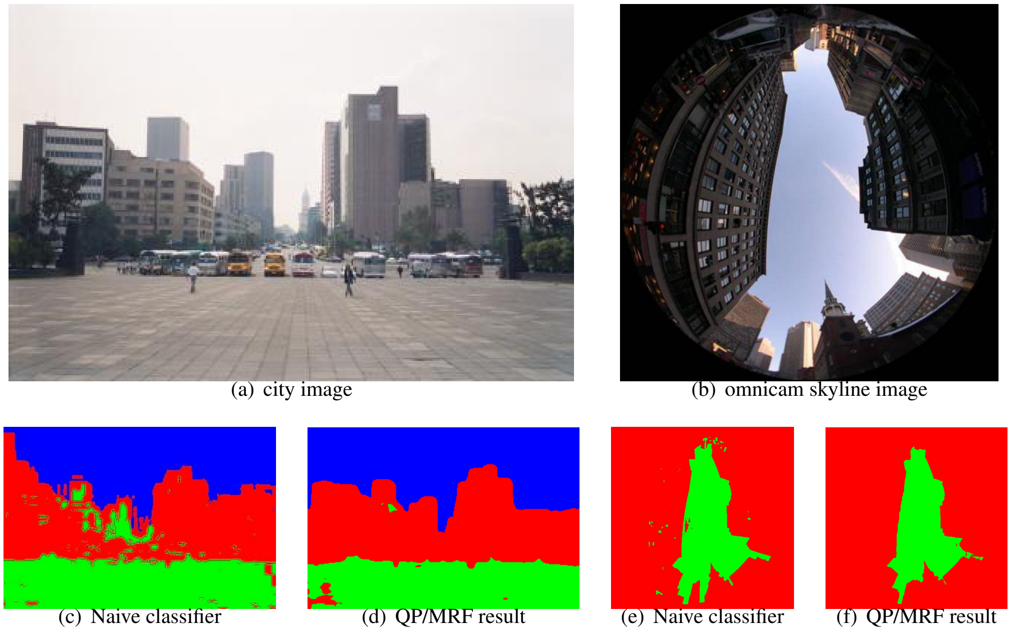
Image super-resolution (SR) refers the process of combining a set of low resolution images into a single high resolution image. Each low-res image g_k is assumed to be generated from an ideal high-res image f via a displacement S_k, a decimation D_k, and a noise process n_k:

$$g_k = D_k S_k f + n_k, \quad k = 1, 2, \cdots, K \tag{6}$$

We use the registration algorithm of Chung, Haber and Nagy [9] to estimate the displacement matrices S_k. Then we reconstruct the high-resolution image by iteratively solving the NNLS

$$\min_{f \geq 0} \frac{1}{2} \sum_{k=1}^{K} ||D_k S_k f - g_k||^2.$$

Our test dataset is taken by Multi-Dimensional Signal Processing Research Group (MDSP) (`http://www.soe.ucsc.edu/~milanfar/software/sr-datasets.html`). There are 30 uncompressed grayscale low-resolution images of size 57 × 49, from which we reconstructed high-resolution image with size 285 × 245 (see Figure 3.1). Commercial solvers produced an identical result, albeit two orders of magnitude slower.


(a) city image

(b) omnicam skyline image

(c) Naive classifier

(d) QP/MRF result

(e) Naive classifier

(f) QP/MRF result

Fig. 2. Sample image labeling results. (a,b): Input images. A small number of training pixels are labeled by the user. (c,e): Result from the trained color classifier. (d,f): Result from the MRF quadratic program using the color classifier as input.

3.2. Image Labeling

In Markov random fields (MRF)-based interactive image segmentation techniques, the user labels a small subset of pixels, and the MRF propagates these labels across the image, typically finding high-gradient contours as segmentation boundaries where the labeling changes. The problem is combinatorial if there are more than two kinds of label, and usually approximately optimized by network flow algorithms.

The problem can also be relaxed to a quadratic program: Let r be a pixel in the image or the region of interest $\Omega = \{r_i : i = 0, 1, \cdots, N\}$, the class set is denoted by $\mathscr{K} = \{1, 2, \cdots, K\}$. Then the probabilistic segmentation approaches compute a probability measure field $x = \{x_k(r) : r \in \Omega, k \in \mathscr{K}\}$ such that $\sum_{k=1}^{K} x_k(r) = 1$, $x_k(r) \geq 0$, for all $k \in \mathscr{K}$ and $r \in \Omega$. The set of neighbours pixel r is denoted as $\mathscr{N}_r = \{s \in \Omega : |r - s| = 1\}$. The QP has the form

$$\begin{aligned} \min_{x} \quad &= \quad \sum_{k=1}^{K} \sum_{r} \left(\frac{\eta}{2} \sum_{s \in \mathscr{N}(r)} \omega_{rs} (x_k(r) - x_k(s))^2 + d_{rk} x_k(r) \right), \\ \text{s.t.} \quad & \quad \sum_{k=1}^{K} x_k(r) = 1, \quad x_k(r) \geq 0, \quad \forall k \in \mathscr{K}, \forall r \in \Omega \qquad (7) \end{aligned}$$

where d_{rk}—the cost of assigning label k to pixel r—is obtained from a simple color classifier that is trained from a small set of pixels labeled by the user. For this we used an SVM, itself trainable using multiplicative update rule in eq. (5). The quadratic term controls the granularity of the regions, i.e. promotes smooth regions. The spatial smoothness is globally controlled by the positive parameter η and locally controlled by weight ω_{rs}, which is chosen such that $\omega_{rs} \approx 1$ if the neighbouring pixels r and s are likely to belong to the same class and $\omega_{rs} \approx 0$ otherwise. In our experiments, we set $\omega_{rs} = \langle I_r, I_s \rangle / (|I_r||I_s|)$, the cosine of the angle between two pixels' vectors in LAB color space.

This QP is a convex relaxation of a combinatorial {0,1}-labeling problem. A somewhat more complicated QP is presented in [10]; it has extra bias terms to make the QP strictly convex, as needed by their solution algorithm.

One can optimize either QP using Eq. (1) to solve its dual, or one can derive multiplicative update rules directly from the Lagrangian using the technique demonstrated in section 1. Here the first-order KKT optimality conditions

$$\begin{aligned} \eta \sum_{s \in \mathscr{N}(r)} \omega_{rs} (x_k(r) - x_k(s)) + d_{rk} - \lambda_r &= 0 \\ \eta x_k(r) \sum_{s \in \mathscr{N}(r)} \omega_{rs} - \eta \sum_{s \in \mathscr{N}(r)} \omega_{rs} x_k(s) + d_{rk} - \lambda_r &= 0 \end{aligned}$$

yield a two-step update for primal and dual parameters

$$\begin{aligned} x_{rk} &\leftarrow x_k(r) \frac{\eta \sum_{s \in \mathscr{N}(r)} \omega_{rs} x_k(s) + d_{rk}^{-} + \lambda_r}{\eta x_k(r) \left(\sum_{s \in \mathscr{N}(r)} \omega_{rs} \right) + d_{rk}^{+}} \\ \lambda_r &\leftarrow \lambda_r \frac{1}{\sum_k x_k(r)} \end{aligned}$$

Depending on the size of the image and the number of classes, the MRF QP has 1-2 million variables and converges in 5-6 seconds on a CPU. Some example results are shown in Figure 3.2. Due to sparse user labeling, the simple classifier typically assigns many pixels to the wrong class; most of these errors are corrected by the quadratic program.

4. SUMMARY

By expressing the KKT first-order optimality conditions as a ratio, we obtained a multiplicative fixpoint that quickly solves very large quadratic programs on low-cost parallel compute devices. The update converges from all positive initial guesses; we also proved asymptotic linear convergence rate. Although QP problems are ubiquitous in image processing, QP algorithms are infrequently used because the problems are prohibitively large. The multiplicative update given here makes the QP approach viable, with one algorithm solving many problems. Moreover, the algorithm is well suited for SIMD/MIMD parallel processors now being deployed in consumer electronics. We demonstrated applications to super-resolution and image segmentation, and show that other algorithms for deblurring, tomography, and matrix factorization are in fact special cases of the general update rule.

A. PROOF OF LINEAR CONVERGENCE

Proof Let P_i be the ith row of Q^+ and N_i the ith row of Q^-. Consider perturbing the ith element of the optimum x^* with some nonzero value $\varepsilon > -x_i^*$ to yield $x = x^* + \varepsilon e_i$, then applying one update iteration to obtain $\hat{x}$. The ratio of errors between successive iterations is

$$\left|\frac{\hat{x}_i - x_i^*}{x_i - x_i^*}\right| = \left|\frac{\frac{h_i^+ + N_i(x^*+\varepsilon e_i)}{h_i^- + P_i(x^*+\varepsilon e_i)} x_i - x_i^*}{x_i - x_i^*}\right| = \left|\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\frac{x_i}{\varepsilon} - \frac{x_i^*}{\varepsilon}\right|$$

$$\begin{aligned}
&= \left|\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\frac{x_i^* + \varepsilon}{\varepsilon} - \frac{x_i^*}{\varepsilon}\right| \\
&= \left|\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon} + \left(\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon} - 1\right)\frac{x_i^*}{\varepsilon}\right| \\
&= \left|\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon} + \left(\frac{h_i^+ + N_i x^* - P_i x^* - h_i^- - Q_{ii}\varepsilon}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\right)\frac{x_i^*}{\varepsilon}\right| \\
&= \left|\frac{h_i^+ + N_i x^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon} + \frac{h_i - Q_i x^* - Q_{ii}\varepsilon}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\frac{x_i^*}{\varepsilon}\right| \\
&= \left|\frac{h_i^+ + N_i x^* - Q_{ii}x_i^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\right| = \left|\frac{h_i^+ + P_i x^* - Q_i x^* - Q_{ii}x_i^*}{h_i^- + P_i x^* + Q_{ii}\varepsilon}\right|
\end{aligned}$$

To prove a linear rate of convergence, we show that this quantity is less than one. From the first-order Karush-Kuhn-Tucker (KKT) optimality conditions, we have $Qx^* - h \geq 0$. Using that, in the case $h_i \geq 0$ we can continue with

$$\leq \frac{P_i x^* - Q_{ii}x_i^*}{P_i x^* + Q_{ii}\varepsilon} = \frac{(\sum_{i\neq j} P_{ij}x_j^*) + r_i x_i^*}{(\sum_{i\neq j} P_{ij}x_j^*) + r_i x_i^* + Q_{ii}(\varepsilon + x_i^*)}.$$

The numerator and denominator are both non-negative because all terms are non-negative. Note that $Q \succeq 0 \Rightarrow Q_{ii} > 0$, and $\varepsilon > -x_i^*$, so $Q_{ii}(\varepsilon + x_i^*) > 0$ and the denominator is strictly larger than the numerator.

For $h_i < 0$, we use $Q = P - N$ to continue with

$$\begin{aligned}
&= \left|\frac{N_i x^* - Q_{ii}x_i^*}{h_i^- + Q_i x^* + N_i x^* + Q_{ii}\varepsilon}\right| \leq \left|\frac{N_i x^* - Q_{ii}x_i^*}{N_i x^* + Q_{ii}\varepsilon}\right| \\
&= \left|\frac{(\sum_{i\neq j} N_{ij}x_j^*) + r_i x_i^* - Q_{ii}x_i^*}{(\sum_{i\neq j} N_{ij}x_j^*) + r_i x_i^* + Q_{ii}\varepsilon}\right| \qquad (8)
\end{aligned}$$

If $x_i^* = 0$, Eq. (8) is trivially < 1. Otherwise, since $\varepsilon > -x_i^*$ we have $r_i x_i^* + Q_{ii}\varepsilon > r_i x_i^* - Q_{ii}x_i^*$. Simple algebra then shows that Eq. (8) is < 1 for any choice of $r_i \geq Q_{ii} - (\sum_{i\neq j} N_{ij}x_j^*)/x_i^*$, including $r_i = Q_{ii}$. ∎

B. REFERENCES

[1] H.W. Kuhn and A.W. Tucker, "Nonlinear programming," in *Proc. 2nd Berkeley Symposium*, 1951, pp. 481–492.

[2] Matthew Brand, Vijay Shilpiekandula, Chen Yao, and Scott A. Bortoff, "A parallel quadratic programming algorithm for model predictive control," in *Proc. 18th World Congress of the International Federation of Automatic Control (IFAC)*, 2011, p. to appear.

[3] M. E. Daube-Witherspoon and G. Muehllehner, "An iterative image space reconstruction algorthm suitable for volume ECT," *Medical Imaging, IEEE Transactions on*, vol. 5, no. 2, pp. 61–66, june 1986.

[4] D. Lee and S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, T.K. Leen, T.G. Dietterich, and V. Tresp, Eds. April 2001, pp. 556–562, MIT Press.

[5] M. Bertero, H. Lanteri, and L. Zanni, "Iterative image reconstruction: A point of view," in *Proc. Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and IMRT*, 2007.

[6] P. P. B. Eggermont, "Multiplicative iterative algorithms for convex programming," *Linear Algebra and its Applications*, vol. 130, pp. 25–42, 1990.

[7] Z.Q. Luo and P. Tseng, "On the linear convergence of descent methods for convex essentially smooth minimization," *SIAM J. Control and Optimization*, vol. 30, no. 2, pp. 408–425, 1992.

[8] M. Benzi, G.H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 1, pp. 1–137, 2005.

[9] J. Chung, E. Haber, and J. Nagy, "Numerical methods for coupled super-resolution," *Inverse Problems*, vol. 22, pp. 1261–1272, 2006.

[10] M. Rivera, O. Dalmau, and J. Tago, "Image segmentation by convex quadratic programming," 2008, pp. 1–5.