AN EXTENSION OF AN INTERIOR-POINT METHOD FOR ENTROPY MINIMIZATION

Irina F. Gorodnitsky

Cognitive Science Department, UC of San Diego
La Jolla, CA 92093-0515
E-mail: igorodni@cogsci.ucsd.edu

ABSTRACT

Entropy optimization is used in signal compression, coding, estimation, and resource scheduling, among other applications. The paper presents a novel algorithm for entropy optimization. The algorithm is motivated by the efficient interior-point methods developed in Linear Programming. The algorithm uses a *Generalized Affine Scaling Transformation* that is an extension of the *Affine Scaling Transformation* utilized in interior-point methods. I show that for some entropy functions the proposed algorithm has superior convergence properties when compared to comparable the interior-point methods. The proposed algorithm is also shown to be related to, and a more general case of, the recently developed FOCUSS algorithm.

1. INTRODUCTION

The Entropy optimization problem (EOP) occurs in many areas of engineering. Its applications include image recontraction [1], failure diagnosis [2], and finding system's equilibrium [3], among other applications. In signal processing and communication theory, entropy optimization is used for efficient compression, estimation, and de-noising of signals [4, 5, 6]. These problems are sometimes called the best basis selection problem where entropy is used as a measure of concentration. The entropy of a process in this case can have a meaningful statistical interpretation, such as the entropy of the probability density function (pdf).

This paper presents a novel algorithm for solving entropy optimization problems. The development is motivated by the work in interior-point methods in Linear Programming (LP). Linear Programming has undergone revolutionary development in the last 15 years, which has led to the the development of new efficient algorithms. Much of this advance in Linear Programming has been associated with the development of *interior-point methods* (IMPs). Several previously unsolvable large-scale LP problems were solved in the last decade owing to the work on these methods. IPMs for entropy programs have also been developed, but the conversion of theoretical results into numerical algorithms has been slow so far.

The method proposed here differs from the standard IPM. The proposed method uses a *Generalized Affine Scaling Transformation* (GAST) which is a more general form of the *Affine Scaling Transformation* (AST) used in the affine scaling IPMs. GAST leads to a natural simplification of the affine scaling IPM for the EOP. The proposed GAST algorithm, abbreviated as GASTA, is shown to have higher rate of local convergence than the affine scaling interior-point method for certain entropy functions. This algorithm is also shown to be related to the recently developed FOCUSS algorithm [6], with FOCUSS being a special case of GASTA. The work presented here unifies these optimization schemes.

2. PROBLEM DEFINITION AND BACKGROUND

2.1. The Entropy Optimization Problem

The general convex optimization problem with linear constraints is

$$\begin{aligned} \text{minimize} \quad & h(\mathbf{x}) = \sum_{i=1}^{n} h_i(x_i) \qquad (1) \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b}, \end{aligned}$$

$A \in R^{m\times n}$ and $\mathbf{b} \in R^m$. It is called an entropy optimization problem if each h_i is an entropy function defined as

$$h_i''(x_i) = \mu_i(|x_i|^{d_i} + q_i), \qquad (2)$$

where $\mu_i \geq 0, q_i \geq 0$, and $d_i \in R$ are constants. The entropy functions in this definition include $|x|^2$, $-\sqrt{|x|}$, $\frac{1}{|x|}$, $log|x|$, and $|x|log|x|$, among others. This definition allows complex-valued x.

Entropy functions take on small values when all but a few entries of a sequence are negligible and large values otherwise. They are used in many signal processing applications to measure *concentration* or *spatial localization* of a sequence, which, in practical terms, is the number of non-negligible elements in a sequence. Several entropy functions are commonly used in signal compression and estimation and have statistical interpretations. For example,

This work is supported by the NSF grant no. EIA-9806107

$\sum_{i=1}^{n} log(|x_i|) = \sum_{i=1}^{n} log|u_i|^2$ can be interpreted as the entropy of a Gauss-Markov process $k \to u(k)$. Minimizing this function over all bases finds the *Karhunen-Loeve basis* for the process [4]. The function $-|x_i|log|x_i|$ can be interpreted as the entropy of the probability distribution function (*pdf*) of a normalized sequence $|\mathbf{x}|$ [4]. It is also known as the Shannon entropy. This function is actually the negative of the entropy function defined by (2). Other entropy functions used to measure concentration include the l_p family of entropies $||\mathbf{x}||_p^p, p \in [0, 2]$ [5]. Of those, l_0 and l_1 are not defined as entropies under (2), while l_p entropy for $p \in (0, 1)$, must have a negative sign in accordance with (2).

2.2. Interior-point Methods

The notation used here and throughout the rest of the paper follows closely the notation in [7].

Interior-point methods were first developed to efficiently solve LP problems. The standard form of the LP problem is

$$\begin{aligned} \text{minimize} \quad & c^T \mathbf{x} \qquad (3) \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b}, \quad x \geq 0. \end{aligned}$$

The *relative interior* of the LP problem is defined as the interior of the feasible domain

$$P = \{\mathbf{x} \in R^n | A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}. \qquad (4)$$

In LP the notion of *polynomial* performance is used to evaluate algorithms. A method is called polynomial if the number of operations required by this method to solve a given problem is bounded from above by a polynomial in the problem 'size'. This measure provides 'the worst case' bound on computations required by a method.

IPMs were first introduced and analyzed in the 1960s. They were largely ignored until Karmarkar's derivation of a polynomial interior-point algorithm for LP in 1984. The innovative part of his gradient descent algorithm was the use of the *projective scaling transformation* to re-scale the solution after each iteration. The proof of polynomial complexity of the Karmarkar algorithm led an explosion of interest in IPMs, with many new interior-point algorithms developed and the works from the 1960s re-examed.

Although many versions of IPMs exist now, they can be delineated into four categories: 1) Path-following methods; 2) Affine scaling methods; 3) Projective potential reduction methods; and 4) Affine potential reduction methods. The algorithms can be distinguished further as primal-only, dual-only, or primal-dual algorithms within each of the categories. Each subgroup can be divided further into short, medium and long-step algorithms. Although the literature on the IPM is enormous and includes thousands of papers, it can be shown that all IPMs rely on two common concepts. They all utilize a search direction and the central path, which is a smooth curve in the interior of the feasible domain that ends in the optimal solution [8]. The search direction for each IPM can be shown to be a linear combination of two characteristic vectors: the affine scaling and the centering direction [8]. The algebraic path taken by each IPM is different, and which algebraic path is computationally superior remains an unanswered question.

The development of the algorithm here is motivated by the primal affine scaling IPM, first proposed by I.I.Dikin in 1967 and later rediscovered in 1986 by Vanderbei at al. and by Barnes. This method turned out to be a natural simplification of Karmarkar's projective algorithm. The primal and dual affine scaling methods have proved to be efficient in practice. This observation was in large part responsible for the initial surge in interest in interior-point methods.

The primal affine scaling method is based on two operations. First, an *Affine Scaling Transformation* (AST) is carried out to 'center' the current iterate, i.e. transform it into the vector $\mathbf{e} = (1, \cdots, 1)^T$. The second operation is a step in the steepest descent direction in the null space of the transformed linear constraints.

Mathematically, this is described as follows. Assume $\mathbf{x}_k$ is a feasible solution to the system $A\mathbf{x} = \mathbf{b}$. Define an $n \times n$ diagonal matrix

$$X_k = diag(\mathbf{x}_k). \qquad (5)$$

The Affine Scaling Transformation is defined as

$$\mathbf{y} = T_k(\mathbf{x}) = X_k^{-1}\mathbf{x}. \qquad (6)$$

This transformation rescales each of the components of $\mathbf{x}$. The problem in the new variable $\mathbf{y}$ becomes

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}_k^T \mathbf{y} \qquad (7) \\ \text{subject to} \quad & A_k \mathbf{y} = \mathbf{b}, \qquad (8) \end{aligned}$$

where $A_k = AX_k$ and $\mathbf{c}_k = X_k\mathbf{c}$. The steepest descent direction is found by projecting the negative gradient $-\mathbf{c}_k$ into the null space of A_k:

$$P_k(-\mathbf{c}_k) = (I - A_k^+ A_k)(-\mathbf{c}_k), \qquad (9)$$

The updated solution in the transformed space is

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \alpha_k P_k(-\mathbf{c}_k), \qquad (10)$$

for some step length α_k. The update in the original space is recovered from $\mathbf{y}_{k+1}$ using the inverse AST.

The affine scaling method was generalized to convex programming in [9]. The approach taken here is different, however. First, to make the method easy to apply to enginering problems, $\mathbf{x}$ is not restricted to be positive, and can be negative or complex-valued. Second, a *Generalized* AST (GAST) is proposed and used in place of the AST to scale the variables. The resulting GAST algorithm (GASTA) is simpler computationally than the affine scaling algorithm and can be seen as its natural simplification. Further, in

IPMs, the steps are restricted to the interior of the feasible domain. Here, this restriction is removed and the only requirement is that the objective function is reduced at each step. GAST systematically re-scales the components of $\mathbf{x}$ after each iteration. In summary, the affine scaling method is interpreted quite broadly in the development undertaken here. The feature of the affine scaling method that is used is the search direction, which is a combination of a scaling and the steepest descent vector, with the scaling being generalized. In fact, the GAST algorithm is not necessary an IPM. But because its search direction is closely related to that of an IPM, I use the term extended IPM to indicate this relation and to emphasize the continuity of this development within other work in functional optimization.

Although the affine scaling methods proved to be efficient in practice, many theoretical questions concerning convergence and performance of these methods remain unanswered. Only global convergence under some conditions has been proven for these algorithms, although it was proven for convex programs under a primal non-degeneracy assumption in [9]. Further, these algorithm are believed not to be polynomial, although this has not been proven conclusively.

The theoretical results presented here show that the GAST algorithm has some desirable convergence properties. Its rate of convergence is shown to be superior to that of affine scaling methods for certain entropies. These results provide encouragement that GAST may be useful for solving certain EOPs. The results also give insight into how this approach may be extended further to design more efficient algorithms for EOPs.

3. AN EXTENSION OF AFFINE SCALING METHOD FOR ENTROPY MINIMIZATION

For simplicity and without loss of generality, assume that

$$h_i''(x_i) = |x_i|^d, \quad \forall \ i. \tag{11}$$

For now, let's further assume that $h_i'(x_i) = \frac{1}{d+1}|x|_i^{d+1}$. This means that $h(x)$ contains no linear cost components nor products involving $\log|x|$. Note that this eliminates the Shannon entropy function $|x|\log|x|$ but permits $\log|x|$. The reason for the second assumption is that we want to illuminate certain properties of the algorithm which is easier to do in the simplified presentation. The general result for entropies in (11) is given at the end.

Define GAST as

$$\begin{aligned} S_k &= diag(|\mathbf{x}_k|^{-\frac{d}{2}}) \\ \mathbf{y} &= T_k(\mathbf{x}) = S_k^{-1}\mathbf{x}, \end{aligned} \tag{12}$$

where d is the same as in (11).

The problem in the new variable y becomes

$$\begin{aligned} \text{minimize} \quad & h_k(\mathbf{y}) \\ \text{subject to} \quad & A_k\mathbf{y} = \mathbf{b}, \end{aligned} \tag{13}$$

where $A_k = AS_k$ and $h_k(\mathbf{y}) = h(S_k\mathbf{y})$ is the transformed objective.

GAST does not 'center' the solution $\mathbf{x}_k$ to a point equidistant from all the axes, as does the AST. Instead one can think of it as a natural scaling for the objective $h(\mathbf{x})$.

Steepest descent direction is still the best choice for reducing the objective cost, since the Hessian of the cost is diagonal here. The gradient with respect to $\mathbf{y}$ of the scaled objective function is

$$\nabla_y h(S_k\mathbf{y}) = S_k \nabla_x h(\mathbf{x}) = S_k\frac{1}{d+1}X^{d+1}\mathbf{e} = \\ \frac{1}{d+1}S_kX^d\mathbf{x}\Big|_{\mathbf{x}_k} = \frac{1}{d+1}S_kX_k^dS_k\mathbf{y}_k = \frac{1}{d+1}\mathbf{y}_k,$$

where $\mathbf{y}_k = S_k^{-1}\mathbf{x}_k$ is the current interior solution in the transformed space. The motivation for the choice of GAST should become clear now. The projected gradient direction $\mathbf{d}_k^y$ in this case is the projection of only the scaled solution itself:

$$\mathbf{d}_k^y = P_k(\frac{1}{d+1}\mathbf{y}_k) = \frac{1}{d+1}[I - A_k^+A_k]\mathbf{y}_k, \tag{14}$$

and the new solution $\mathbf{y}_{k+1}$ in the transformed space is

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \frac{\alpha_k}{d+1}[I - A_k^+A_k]\mathbf{y}_k = \mathbf{y}_k - \frac{\alpha_k}{d+1}(\mathbf{y}_k - A_k^+\mathbf{b}). \tag{15}$$

The solution in the original space is recovered as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k(\mathbf{x}_k - S_kA_k^+\mathbf{b}), \tag{16}$$

with $\lambda_k = \frac{\alpha_k}{d+1}$. The moving direction in the transformed space is $\mathbf{y}_k - A_k^+\mathbf{b}$ and the step length is λ_k.

Equation (15) gives a general GAST update equation for an arbitrary step length λ_k. If we now choose $\alpha_k = d+1$, then (15) simplifies further to

$$\mathbf{y}_{k+1} = A_k^+\mathbf{b}, \tag{17}$$

and

$$\mathbf{x}_{k+1} = S_kA_k^+\mathbf{b}. \tag{18}$$

(18) is equivalent to the update step of the FOCUSS algorithm which was in developed in [6]. Equations (12) and (18) define the general FOCUSS algorithm of [6]. FOCUSS was developed to optimize a subset of the entropy functions discussed here and its derivation was not based on the affine scaling method approach. Rather, it was derived as a 'reweighted minimum norm algorithm' where the solution of the preceding iteration acted as a prior to constrain the minimum norm solution in the next iteration. The exact connection of FOCUSS to the primal affine scaling method has not been made clear until now.

For the two of most common families of entropy measures $h(\mathbf{x}) = -\sum_i^n |x(i)|^p$, $p \in (0,1)$ and the Gaussian entropy $h(\mathbf{x}) = \sum_i^n ln|x_i|$, the GASTs respectively become

$$S_k = diag|\mathbf{x}_k|^{\frac{2-p}{2}} \tag{19}$$

and

$$S_k = diag|\mathbf{x}_k|. \tag{20}$$

Note that the GAST for $h(\mathbf{x}) = \sum_i^n ln|x_i|$ is equivalent to AST and the GAST algorithm in this case is equivalent to the primal affine scaling method for $x_i > 0$.

The notion of polynomiality that is used in LP to evaluate complexity of an algorithm is not defined for concave functions, so computational complexity must be treated in a different manner here. Here I discuss local and global convergence of GASTA.

Global Convergence of GASTA: In [6], FOCUSS was shown to converge globally for S_k given by (19) for integer $p < 1$. S_k in (20) becomes a special case of (19). Following the basic steps of this proof, it can be extended to show that GASTA is globally convergent for $d < -1$ in (11).

Local Convergence of GASTA: Here again we can refer to the local convergence results of FOCUSS in [6] to show that the local rate of convergence for GASTA is $|d|$. This means that GASTA is quadratically convergent for the Gaussian entropy $h(\mathbf{x}) = \sum_i^n ln|x_i|$ and its convergence is superquadratic for $h(\mathbf{x}) = \sum_i^n |x(i)|^p$ for $p < 0$. Note that the best rate of convergence obtained with interior-point methods for convex programming is quadratic. This means that GASTA has better convergence rate for entropy functions $h(\mathbf{x}) = \sum_i^n |x(i)|^p, p < 0$, and the rate becomes larger with the decrease in p. This result is very evident in practice. It typically takes less than 9 iterations for GASTA to converge to a solution when $p < -1$ is used, almost regardless of the problem size.

GAST rule for the general entropy function: Following the above derivation, we can recognize the GAST for the general entropy function (11) to be:

$$S_k = diag\left(\frac{h_i'(x_i)}{x_i}\right)^{-1/2}. \tag{21}$$

4. DISCUSSION

It is important to realize that sophisticated implementation techniques play the key role in the claimed efficiency of interior point methods. Existence of theoretical results does not automatically translates into a valuable algorithm in practice. For example, in the case of interior-point methods for convex programming, the conversion of theoretical results into numerical algorithms has been very slow.

This makes the connection presented here, of GASTA to the IPM, even more important because it opens up the opportunity of using the existing results in IPM for implementation of the GAST algorithm. Implementation issues include the initialization of the algorithm, checking for optimality, minimizing the computational complexity, and regularization of the algorithm to deal with noisy data.

For example, the computational bottleneck of both GASTA and IPMs is in inverting a matrix to find the descent direction. Here one can and should rely on the existing implementations of Cholesky, CG, or LQ factorizations, including the use of good sparse matrix techniques, developed for the IPM. Similarly, the regularization methods developed for the IPM can be tapped into for regularization of GASTA.

Finally, I would like to comment on some of the questions that are opened up by this work. What GASTA shows is that a departure from the strict affine scaling transformation may be beneficial for optimizing at least some of the objective functions. This brings up the question of whether the 'best' scaling can be determined as a function of the optimized cost itself. Another question this brings up is whether other IPMs for convex optimization may be extended based on the ideas discussed here. For example, the path-following IPMs are thought to be the most promising of all the IPMs for convex programming, so further extension of these methods may prove beneficial.

5. REFERENCES

[1] G.T. Herman. A relaxation method for reconstructing objects from noisy x-rays. *Math. Programming*, 8:1–19, 1975.

[2] P. Duhamel and J.C. Raut. Automatic test generation techniques for analog circuits and systems: a review. *IEEE Trans. Circuits and Systems*, Jul. 1979.

[3] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

[4] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best-basis selection. *IEEE Trans. Information Theory*, 38:713–718, 1992.

[5] D. Donoho. On minimum entropy segmentation. In C. K. Chui et al., editor, *Wavelets: Theory, Algorithms, and Applications*, pages 233–269. Academic Press, 1994.

[6] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: A recursive weighted minimum norm algorithm. *IEEE Trans. Sig. Process.*, 45(3):600–616, Mar. 1997.

[7] S-C. Fang and S. Putenpura. *Linear Optimization and Extensions.* Prentice Hall, Engelwood Cliffs, NJ, 1993.

[8] D. Den Hertog and C. Roos. A survey of search directions in interior point methods for linear programming. *Mathematical Programming*, 52:481–509, 1991.

[9] C.C. Gonzaga and L.A. Carlos. A primal affine-scaling algorithm for linearly constrained convex programs. *Report ES-238/90*, Dept. of Systems Eng. and Comp. Sci., Univ. of Rio de Janeiro, 1990.