Sparse Component Analysis in Presence of Noise Using an Iterative EM-MAP Algorithm

Hadi Zayyani[1], Massoud Babaie-Zadeh[1,⋆], G. Hosein Mohimani[1], and Christian Jutten[2]

[1] Electrical Engineering Department, Advanced Communications Research Institute (ACRI), Sharif University of Technology, Tehran, Iran
[2] GIPSA-lab, Department of Images and Signals, National Polytechnic Institute of Grenoble (INPG), France
zayyani2000@yahoo.com, mbzadeh@yahoo.com, gh1985im@yahoo.com, Christian.Jutten@inpg.fr

Abstract. In this paper, a new algorithm for source recovery in under-determined Sparse Component Analysis (SCA) or atomic decomposition on over-complete dictionaries is presented in the noisy case. The algorithm is essentially a method for obtaining sufficiently sparse solutions of under-determined systems of linear equations with additive Gaussian noise. The method is based on iterative Expectation-Maximization of a Maximum A Posteriori estimation of sources (EM-MAP) and a new steepest-descent method is introduced for the optimization in the M-step. The solution obtained by the proposed algorithm is compared to the minimum ℓ^1-norm solution achieved by Linear Programming (LP). It is experimentally shown that the proposed algorithm is about one order of magnitude faster than the interior-point LP method, while providing better accuracy.

Keywords: sparse component analysis, sparse decomposition, blind source separation, independent component analysis.

1 Introduction

Finding (sufficiently) sparse solutions of under-determined systems of linear equations (possibly in the noisy case) has been studied extensively in recent years [1,2,3,4,5,6,7,8,9,10]. The problem has a growing range of applications in signal processing. One of these applications is the noisy under-determined sparse source separation which is also called Sparse Component Analysis (SCA) [1, 2, 3, 4, 5, 6]. Another application is the so-called 'atomic decomposition' problem which aims at finding a sparse representation for a signal in an overcomplete dictionary [7,8,9,10]. In this paper, we will mainly use the context of SCA stating our approach. The discussions, however, may be easily followed in other contexts of application such as atomic decomposition.

SCA can be viewed as a method to achieve separation of sparse sources. The Blind Source Separation (BSS) problem is to recover m unknown sources from n

⋆ This work has been partially funded by Sharif University of Technology, by Center for International Research and Collaboration (ISMO) and by Iran NSF (INSF).

M.E. Davies et al. (Eds.): ICA 2007, LNCS 4666, pp. 438–445, 2007.

observed mixtures of them, where little or no information is available about the sources (except their statistical independence) and the mixing system. In this paper we consider the noisy linear instantaneous model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t). \tag{1}$$

where $\mathbf{x}(t)$, $\mathbf{s}(t)$ and $\mathbf{n}(t)$ are $n \times 1$, $m \times 1$ and $n \times 1$ vectors of sources, mixtures and white Gaussian noises, respectively, and $\mathbf{A}$ is the $n \times m$ mixing matrix. In the under-determined case ($m > n$), estimating the mixing matrix is not sufficient to recover the sources, since the mixing matrix is not invertible. Then, the estimation of sources requires some prior information on the sources and passes from a blind problem to a semi-blind problem. One such prior information is the sparsity of sources. It means that only a few samples of the sources are nonzero (say they are active) and most of them are almost zero (say they are inactive).

Then SCA can be solved in two steps: first estimating the mixing matrix, and then estimating the sources. The first step may be accomplished by means of clustering [1] or other methods [4]. The second step requires finding the sparse solution of (1) assuming $\mathbf{A}$ to be known [9]. In this paper, we focus on the source estimation assuming $\mathbf{A}$ is known.

In the atomic decomposition viewpoint [10], we have one signal whose samples are collected in the $m \times 1$ signal vector $\mathbf{s}$ and the objective is to express it as a linear combination of a set of predetermined signals where their samples are collected in vector $\{\varphi_i\}_{i=1}^m$. After [11], the φ_i's are called atoms and they collectively form a dictionary over which the signal is to be decomposed. In this paper, we also consider a noise term for the decomposition. So we can write $\mathbf{s} = \sum_{i=1}^m \alpha_i \varphi_i = \mathbf{\Phi}\alpha + \mathbf{n}$, where $\mathbf{\Phi}$ is the $n \times m$ dictionary (matrix) where the columns are the atoms and α is the $m \times 1$ vector of coefficients. The vector $\mathbf{n}$ can be interpreted as either the noisy term of the original signal that we intend to decompose or the allowed error for the decomposition process.

To obtain the sparse solution of (1), an approach is to search solutions having minimal ℓ^0 norm, *i.e.*, minimum number of nonzero components. This method is intractable when the dimension increases (due to combinatorial search), and it is too sensitive to noise (due to discontinuity of ℓ^0 norm). One of the most successful approaches is Basis Pursuit (BP) [10] which finds the minimum ℓ^1 norm of (1) which can be easily implemented by Linear Programming (LP) methods (especially fast interior-point LP solvers). Another approach is Matching Pursuit (MP) [11] which is very fast, but is somewhat heuristic and does not provide good estimation of sources.

In [5], we proposed a three step (sub-)optimum (in MAP sense) method for SCA in the noisy under-determined case (briefly called MAP) which has the drawback of great complexity and is not tractable for sparse decomposition application (which requires large values of m and n). This problem exists also in [6]. In this article, we propose an iterative method to tackle the great complexity of our MAP method. In the maximization step of our algorithm, we propose here an optimization method based on steepest-descent. Our method results in a fast

sparse decomposition (faster than interior point LP) while improving the quality of source recovery because of its optimality in the MAP sense and dealing with noise.

2 System Model

The noise vector in the model (1) is assumed zero-mean Gaussian with covariance matrix $\sigma_n^2\mathbf{I}$. For modeling the sparse sources the following model is used: the sources are inactive with probability p, and are active with probability $1-p$ (sparsity of sources implies that p should be near 1). In the inactive case the sample of sources is zero and in the active case the sample has a Gaussian distribution. We call this model the 'spiky model' which is a special case of the Bernoulli-Gaussian model used in [5]. The probability density function (PDF) of the sources is then:

$$p(s_i) = p\delta(s_i) + (1-p)N(0,\sigma_r^2). \tag{2}$$

In this model, any sample of the sources can be written as $s_i = q_i r_i$ where q_i is a binary variable (with binomial distribution) and r_i is the amplitude of i'th source with Gaussian distribution. So the source vector can be written as:

$$\mathbf{s} = \mathbf{Q}\mathbf{r} \qquad \mathbf{Q} = \mathrm{diag}(\mathbf{q}). \tag{3}$$

We refer the vector $\mathbf{q} \triangleq [q_1, \ldots, q_m]'$ as the 'source activity vector', where $'$ denotes vector/matrix transposition. Each element of this vector shows the activity of the corresponding source. That is:

$$q_i = \begin{cases} 1 \text{ if } s_i \text{ is active with probability } p \\ 0 \text{ if } s_i \text{ is inactive with probability } 1-p \end{cases} \tag{4}$$

The probability of source activity vector $p(\mathbf{q})$ is equal to:

$$p(\mathbf{q}) = (1-p)^{n_a}(p)^{m-n_a}. \tag{5}$$

where n_a is the number of active sources or the number of 1's in $\mathbf{q}$.

3 Review of Our Previous MAP Algorithm

In [5] we proposed a three step MAP algorithm for the noisy sparse component analysis. The parameter estimation step is done by a novel method based on second and fourth order moments of one mixture and an EM algorithm. The source activity estimation step is done with a MAP method that maximizes the posterior probability. This step is the maximization of:

$$p(\mathbf{q})p(\mathbf{x}|\mathbf{q}) = \frac{p(\mathbf{q})}{\sqrt{det(2\pi\mathbf{Q}_q)}} \exp(\frac{-1}{2}\mathbf{x}'\mathbf{Q}_q^{-1}\mathbf{x}). \tag{6}$$

where $\mathbf{Q}_q = \mathbf{A}\mathbf{V}_q\mathbf{A}' + \sigma_n^2\mathbf{I}$, in which, $\mathbf{Q}_q \triangleq E\{\mathbf{x}\mathbf{x}' \mid \mathbf{q}\}$ and $\mathbf{V}_q \triangleq E\{\mathbf{s}\mathbf{s}' \mid \mathbf{q}\} = \sigma_r^2\mathbf{Q}$ are the conditional covariances of observations and sources given $\mathbf{q}$. After finding the optimum source activity vector, the source amplitudes are estimated as:

$$\widehat{\mathbf{r}} = \sigma_r^2\mathbf{Q}\mathbf{A}'(\sigma_r^2\mathbf{A}\mathbf{Q}\mathbf{A}' + \sigma_n^2\mathbf{I})^{-1}\mathbf{x}. \tag{7}$$

Maximization of (6) is done over discrete space of vector $\mathbf{q}$ with 2^m discrete elements. In [5] this maximization had been done through an exhaustive search on all these 2^m cases.

In this paper, this maximization is done by first converting it to a continuous maximization and then to use a steepest descent algorithm (this is similar to the idea used in [9]). To convert our discrete problem to a continuous one, we use a Mixture of two Gaussians model centered around 0 and 1 with sufficient small variances. By this method our discrete binomial variable q_i is converted to a continuous variable. To avoid falling into local maxima of (6) a gradually decreasing variance can be used in the different iterations (similar to simulated annealing methods). But (6) is still very complex to derive for providing an efficient optimization method such as steepest-descent.

4 The Iterative EM-MAP Algorithm

The main idea of our algorithm is that the source estimation is equal to estimation of vectors $\mathbf{q}$ and $\mathbf{r}$, as observed from (3). Estimation of $\mathbf{q}$ and $\mathbf{r}$ can be done iteratively. First, an estimated vector $\widehat{\mathbf{q}}$ is assumed and then the MAP estimate of vector $\mathbf{r}$ based on the known estimated vector $\widehat{\mathbf{q}}$ and the observation vector $\mathbf{x}$ is obtained (we refer to it as $\widehat{\mathbf{r}}$). Secondly, the MAP estimate of vector $\mathbf{q}$ is obtained based on the estimated vector $\widehat{\mathbf{r}}$ and the observation vector $\mathbf{x}$ (we refer to it as vector $\widehat{\mathbf{q}}$). Therefore, the MAP estimation of sources is done in two other MAP estimation steps.

In the first step a source activity vector $\widehat{\mathbf{q}}$ is assumed and the estimation of $\mathbf{r}$ will be computed. Because the vector $\mathbf{r}$ is Gaussian, its MAP estimation is equal to the Linear Least Square (LLS) estimation [13] and can be computed as follows:

$$\widehat{\mathbf{r}}_{\mathrm{MAP}} = \widehat{\mathbf{r}}_{\mathrm{LLS}} = E(\mathbf{r}|\mathbf{x}, \widehat{\mathbf{q}}) = E(\mathbf{r}\mathbf{x}'|\widehat{\mathbf{q}})E(\mathbf{x}\mathbf{x}'|\widehat{\mathbf{q}})^{-1}\mathbf{x}. \tag{8}$$

This step can be nominated as Expectation step or Estimation step (E-step). Computation and simplification of (8) (like what done in [5]) leads to the following equation which is similar to (7).

$$\widehat{\mathbf{r}} = \sigma_r^2\widehat{\mathbf{Q}}\mathbf{A}'(\sigma_r^2\mathbf{A}\widehat{\mathbf{Q}}\mathbf{A}' + \sigma_n^2\mathbf{I})^{-1}\mathbf{x}. \tag{9}$$

In the second step we estimate $\mathbf{q}$ based on the known $\widehat{\mathbf{r}}$ and the observed $\mathbf{x}$. The MAP estimation is:

$$\widehat{\mathbf{q}}_{\mathrm{MAP}} = \underset{\mathbf{q}}{\operatorname{argmax}}\, p(\mathbf{q}|\mathbf{x}, \widehat{\mathbf{r}}) \equiv p(\mathbf{q}|\widehat{\mathbf{r}})p(\mathbf{x}|\mathbf{q}, \widehat{\mathbf{r}}) \equiv p(\mathbf{q})p(\mathbf{x}|\mathbf{q}, \widehat{\mathbf{r}}). \tag{10}$$

In (10), $p(\mathbf{q})$ can be computed as a continuous variable:

$$p(\mathbf{q}) = \prod_{i=1}^{m} p(q_i) = \prod_{i=1}^{m} [p \exp(\frac{-q_i^2}{2\sigma_0^2}) + (1-p)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})]. \quad (11)$$

Also the term $p(\mathbf{x}|q, \widehat{\mathbf{r}})$ in (10) can be computed as:

$$p(\mathbf{x}|\mathbf{q}, \widehat{\mathbf{r}}) = p_n(\mathbf{x} - \mathbf{AQ}\widehat{\mathbf{r}}) = (2\pi\sigma_n^2)^{\frac{-m}{2}} \exp(\frac{-1}{2\sigma_n^2}(\mathbf{x} - \mathbf{AQ}\widehat{\mathbf{r}})'(\mathbf{x} - \mathbf{AQ}\widehat{\mathbf{r}})) \ . \quad (12)$$

The second step can be called Maximization step (M-step). The maximization can be done over the logarithm of (10). So this step can be simplified as:

$$M - step: \quad \widehat{\mathbf{q}} = \max_{\mathbf{q}} L(\mathbf{q}). \quad (13)$$

where

$$L(\mathbf{q}) = \sum_{i=1}^{m} \log(p(q_i)) + \frac{-1}{2\sigma_n^2}(\mathbf{x} - \mathbf{AQ}\widehat{\mathbf{r}})'(\mathbf{x} - \mathbf{AQ}\widehat{\mathbf{r}}) \ . \quad (14)$$

Maximization of $L(\mathbf{q})$ in the M-step can be done with the steepest descent method. The main steepest descent iteration is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \mu \frac{\partial L(\mathbf{q})}{\partial \mathbf{q}}. \quad (15)$$

In the appendix, we show that the steepest descent algorithm for the M-step is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \frac{\mu}{\sigma_0^2}\mathbf{g}(\mathbf{q}) + \frac{\mu}{\sigma_n^2} Diag(\mathbf{A}'\mathbf{AQ}\widehat{\mathbf{r}} - \mathbf{A}'\mathbf{x}).\widehat{\mathbf{r}}. \quad (16)$$

where $\mathbf{g}(\mathbf{q})$ is defined in the appendix. In the successive iterations, we gradually decrease the variance σ_0 in the form $\sigma_0^{(i)} = \alpha\sigma_0^{(i-1)}$ where α is selected between 0.6 and 1. Also, the step-size μ should be decreasing, *i.e.*, for smaller σ's, smaller μ's should be applied. This is because for smaller variances, our function under maximization is more fluctuating. So the step size can be decreased in the similar form as $\mu^{(i)} = \alpha\mu^{(i-1)}$. Our simulations show that for $\alpha = .8$ only about 4 or 5 iterations are sufficient to maximize the expression $L(\mathbf{q})$ in the M-step. Also the EM-step converges at the third or fourth iteration. The first initialization of the EM-MAP method is done with the minimum ℓ^2 norm solution.

As we see from (16) the second summand is responsible for increasing the prior probability $p(\mathbf{q})$ while the third summand is responsible for decreasing the noise power $||\mathbf{x} - \mathbf{As}||$. When σ_0 is much larger than σ_n, the second term is more effective than the third term and as a result exactness of $\mathbf{x} = \mathbf{As}$ is more important than sparsity of $\mathbf{s}$. When σ_0 is decreased to be comparable to σ_n, both terms are effective to yield the equilibrium point between sparsity and noise.

In summary, the overall algorithm is an iterative two step (E-step and M-step in (9) and (13) respectively) algorithm in which the M-step is done iteratively with the steepest descent method in (16).

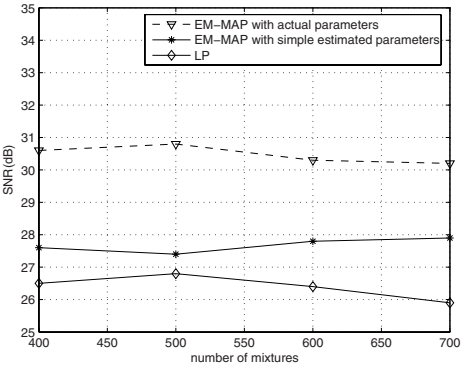

Fig. 1. Comparison of the results of our algorithm in two cases and the (interior-point) LP method. The parameters of simulation are $m = 1000$, $p = .9$, $\sigma_r = 1$, $\sigma_n = .01$, $\alpha = .8$, $\sigma^{(0)} = \widehat{\sigma_r}$ and $\mu^{(0)} = 10^{-6}$. Four iterations are used for EM-step and five iterations for the M-step (steepest descent).

5 Simulation Results

In this section, we examine the performance of our algorithm in two cases (the case of using actual parameters and the case of using a simple method for estimating the parameters which is explained below) and then compare it to the interior-point LP method. Our performance criterion is Signal to Noise Ration (SNR) in dB defined by $\text{SNR} = 10\log_{10}\{\|\mathbf{s}\|^2/\|\hat{\mathbf{s}} - \mathbf{s}\|^2\}$. The simulations have been repeated 50 times (with the same parameters, but for different randomly generated sources and mixing matrix) and the resulting SNR's (in dB) have been averaged.

The values used for the experiment are $m = 1000, n = 400, ..., 700, p = .9, \sigma_r = 1$ and $\sigma_n = .01$. The elements of the mixing matrix are randomly chosen between -1 and 1 and each column normalized to have unit length. In the M-step the value of α is between 0.6 and 1. This parameter effects on the speed of convergence. We use an average value of $\alpha = .8$ in our simulations. The initial value of σ_0 is selected equal to estimated σ_r. The initial value of μ can be selected between 10^{-3} and 10^{-8}. But for small values and large values in this range, the performance is somewhat deteriorated. So we select the value of $\mu = 10^{-6}$. Four iterations are used for the EM-step and five iterations are used for the M-step (steepest descent).

In one of our simulation we use a very simple estimation of the parameters. In this case the parameter p underestimated as $\widehat{p} = .8$. With this assumption and by considering the ergodicity of sources (*i.e.* the mixtures are the ensembles of a random variable $x_j = \sum_{i=1}^{m} a_{ji}s_i + e_j$ where $a_{ji} = b_{ji}/\sqrt{b_{1i}^2 + b_{2i}^2 + ... + b_{ni}^2}$ and b_{ji} is a random variable with uniform distribution on [-1,1] and s_i and e_j are random variables), and by neglecting the noise power, we have $E(x_j^2) = mE(a_{ji}^2)E(s_i^2)$. We know that $\sum_{j=1}^{n} a_{ji}^2 = 1$ (which come from $\sum_{j=1}^{n} a_{ji}^2 = 1$ and the independence of a_{ji}'s), and $E(s_i^2) = (1-p)\sigma_r^2$. With the assumption of $\widehat{p} = .8$, we will have $\widehat{\sigma_r} = \sqrt{\frac{5nE(x_j^2)}{m}}$. For the noise variance, we choose $\widehat{\sigma_n} = \widehat{\sigma_r}/10$.

The results of our simulation are shown in Fig. 5. These results show 3-4 dB improvement (with actual parameters) and 1-2 dB improvement (with simply estimated parameters) of our algorithm over the LP method.

Although, the CPU time is not an exact measure of complexity, it can give us a rough estimation of it, and we compare our algorithm with LP using this measure. Our simulations were performed in MATLAB 7.0 environment using an Intel 2.40 GHz processor with 512 MB of RAM and under Microsoft Windows XP operating system. For one typical simulation, our algorithm takes about 34 seconds while the simulation time of the (interior-point) LP method requires about 204 seconds. So our algorithm is roughly one order of magnitude faster.

6 Conclusions

In this paper, a relatively fast method for finding sparse solution of an underdetermined system of linear equations was proposed. The method was based on the iterative MAP estimation of the sources. This algorithm is approximately one order of magnitude faster than (interior-point) LP, while providing 1-2 dB improvement (with simply estimated parameters). The better performance is obtained due to the optimality of our algorithm which is based on optimum MAP estimation of sources. The simplicity of our algorithm (and its high speed) is obtained due to iterative estimation of source activities and amplitudes and also utilizing an efficient steepest descent for the M-step.

References

1. Zibulevsky, M., Pearlmutter, B.A.: Blind source separation by sparse decomposition in a signal dictionary. Neural Computation 13(4), 863–882 (2001)
2. Gribonval, R., Lesage, S.: A survey of sparse component analysis for blind source separation: principles, perspectives, and new challanges. In: Proceeding of ESANN'06, pp. 323–330 (2006)
3. Davies, M., Mitianoudis, N.: Simple mixture model for sparse overcomplete ICA. In: Puntonet, C.G., Prieto, A.G. (eds.) ICA 2004. LNCS, vol. 3195, pp. 35–43. Springer, Heidelberg (2004)
4. Li, Y.Q., Amari, S., Cichocki, A., Ho, D.W.C, Xie, S.: Underdetermined blind source separation based on sparse representation. IEEE Transaction on Signal Processing 54(2), 423–437 (2006)
5. Zayyani, H., Babaie-Zadeh, M., Jutten, C.: Source estimation in noisy sparse component analysis. Accepted in DSP'2007 (2007)
6. Balan, R., Rosca, J.: Source separation using sparse discrete prior models. In: Proceeding of ICASSP'06 (2006)
7. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal ℓ^1 norm is also the sparsest solution. Technical Report (2004)
8. Donoho, D.L., Elad, M., Temlyakov, V.: Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Transaction on Information theory 52(1), 6–18 (2006)
9. Mohimani, G.H, Babaie-Zadeh, M., Jutten, C.: Fast sparse representation based on smoothed ℓ^0 norm. Accepted in ICA'2007 (2007)

10. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20(1), 31–61 (1999)
11. Mallat, S., Zhang, Z.: Matching pursuit with time-frequency dictionaries. IEEE Transaction on Signal Processing 41(12), 3397–3415 (1993)
12. Djafari, A.M.: Bayesian source separation: beyond PCA and ICA. In: Proceeding of ESANN'06 (2006)
13. Anderson, B.D., Moor, J.B.: Optimal filtering, 2nd edn. Prentice Hall, Englewood Cliffs (1979)

Appendix: Steepest Descent Algorithm

From (13), we have:

$$\frac{\partial L(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \sum_{i=1}^{m} \log(p(q_i)) - \frac{1}{2\sigma_n^2} \frac{\partial}{\partial \mathbf{q}} (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})'(\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}) \,. \tag{17}$$

we define $\mathbf{g}(\mathbf{q}) \triangleq -\sigma_0^2 \frac{\partial}{\partial \mathbf{q}} \sum_{i=1}^{m} \log(p(q_i))$ and $\mathbf{n}(\mathbf{q}) \triangleq (\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})'(\mathbf{x} - \mathbf{A}\mathbf{Q}\widehat{\mathbf{r}})$. With these definitions the scalar function $g(q_i)$ and the $\mathbf{n}(\mathbf{q})$ (with omitting the constant terms) can be computed as:

$$g(q_i) = \frac{pq_i \exp(\frac{-q_i^2}{2\sigma_0^2}) + (1-p)(q_i - 1)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})}{p\exp(\frac{-q_i^2}{2\sigma_0^2}) + (1-p)\exp(\frac{-(q_i-1)^2}{2\sigma_0^2})} \,. \tag{18}$$

$$\mathbf{n}(\mathbf{q}) = -2\mathbf{x}'\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} + \widehat{\mathbf{r}}'\mathbf{Q}\mathbf{A}'\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} \,. \tag{19}$$

with the definitions $\mathbf{C} \triangleq \mathbf{A}'\mathbf{A}$ and $\mathbf{n_1}(\mathbf{q}) \triangleq -2\mathbf{x}'\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}}$ and $\mathbf{n_2}(\mathbf{q}) \triangleq \widehat{\mathbf{r}}'\mathbf{Q}\mathbf{C}\mathbf{Q}\widehat{\mathbf{r}}$ we can write:

$$\frac{\partial n_1(\mathbf{q})}{\partial \mathbf{q}} = \mathrm{diag}(-2\mathbf{x}'\mathbf{A}).\widehat{\mathbf{r}} \,. \tag{20}$$

If we define $\mathbf{W} \triangleq \mathbf{Q}\widehat{\mathbf{r}}$ ($m \times 1$ *vector*) then $\mathbf{n}_2(\mathbf{q}) = \mathbf{W}'\mathbf{C}\mathbf{W}$ and so we have:

$$\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial q_i} = \sum_{j=1}^{m} \frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial W_j} \frac{\partial W_j}{\partial q_i} \,. \tag{21}$$

From the vector derivatives, we have $\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial \mathbf{W}} = 2\mathbf{C}\mathbf{W} \triangleq \mathbf{d}$. Also from the definition of $\mathbf{W}$ we have $\frac{\partial W_j}{\partial q_i} = \widehat{r}_i \delta_{ij}$. So (21) is converted to $\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial q_i} = \sum_{j=1}^{m} d_j \widehat{r}_i \delta_{ij} = \widehat{r}_i d_i$. So the vector form of (21) is equal to:

$$\frac{\partial \mathbf{n}_2(\mathbf{q})}{\partial \mathbf{q}} = \mathrm{diag}(\mathbf{d}).\widehat{\mathbf{r}} \,. \tag{22}$$

From (20) and (22) and $\mathbf{n}(\mathbf{q}) = \mathbf{n}_1(\mathbf{q}) + \mathbf{n}_2(\mathbf{q})$ and definitions of vectors $\mathbf{d}$ and $\mathbf{C}$, we can write:

$$\frac{\partial \mathbf{n}(\mathbf{q})}{\partial \mathbf{q}} = 2\mathrm{diag}(\mathbf{A}'\mathbf{A}\mathbf{Q}\widehat{\mathbf{r}} - \mathbf{A}'\mathbf{x}).\widehat{\mathbf{r}} \,. \tag{23}$$

Finally, (23) and (17) and (15) with the definitions of $\mathbf{n}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ yields the steepest descent iteration in (16).