System identification by a generalized interior point algorithm and nonlinear optimization methods considering ARMA model

Mahdi Sojoodi[1]*, Farzad Soleymani[2]

Received:2015/2/23 Accepted:2015/6/17

Abstract: In this paper, we describe our implementation of an interior point algorithm for large scale systems. First we identify system with small and medium methods convex optimization, then we use interior point method for identification. Finally we offer an interior point method that uses nonlinear cost function and see that we achieve a good trade-off between error and CPU time. Actually, in this paper, we are looking for a method that can identify large scale systems with low model order, error and CPU time of solution of simulation. Previous articles didn't check the order of the computed model, and the relationship between the error and CPU time. We assume that the model of our simulation is ARMA. We are going to identify a large scale system and compute the error and CPU time and compare the relationships. Examined data in this paper is related to cutaneous potential recordings of a pregnant woman. These data are pendulous and have a large standard deviation; therefore, it can't be fitted with ordinary curve fittings, so we use the smoothing spline for computing the order of the model. Finally, we checked the influence of the number of data on error and CPU time and order of model.

Keywords: System Identification; Interior point algorithm; Convex Optimization; Large scale stsen

1. Introduction

System Identification is about building mathematical models of dynamical systems from observed input-output signals. There is a very extensive literature on the subject, with many text books, like [1] and [2]. Most of the techniques for system identification have their origins in estimation paradigms from mathematical statistics, and classical methods like Maximum Likelihood (ML) have been important elements in the area. In this article the main ingredients of this state-of-the-art view of System Identification will be reviewed. This theory is well established and is deployed e.g. in the software [3]. The estimates show attractive asymptotic properties and the methodology has been used extensively and successfully. Some problems can however be listed: (1) the selection of model structures (model orders) is not trivial and may compromise the optimality properties, in particular for shorter data records, and (2) the typically non-convex nature of the criteria may cause numerical optimization artifacts (like ending up in non-global, local minima). Therefore there is a current trend to enforce estimation methods based on convex formulations. So recently, alternative techniques, mostly from machine learning and the convex optimization area have emerged. Also these have roots in classical statistical (Bayesian) theory. The main elements of these will also be reviewed here.

In recent years there has been growing interest in convex optimization techniques for system identification and time series modeling. In [4-5] this interest is motivated by the success of convex methods for sparse optimization and rank minimization in signal processing, statistics, and machine learning, and by the development of new classes of algorithms for large-scale, such as interior point method, nonlinear optimization method, Proximal gradient algorithms and Alternating Direction Method of Multipliers. Low-dimensional model structure in identification problems is typically expressed in terms of matrix rank or sparsity of parameters.

In optimization formulations this generally leads to non-convex constraints or objective functions. However, formulations based on convex penalties that indirectly minimize rank or maximize sparsity are often quite effective as heuristics, relaxations, or, in rare cases, exact reformulations. The best known example is 1-norm regularization in sparse optimization, i.e., the use of the 1-norm $\|x\|_1$ in an optimization problems a substitute for the cardinality of a vector x.

This idea has a rich history in statistics, image and signal processing [6-10]. And an extensive mathematical theory has been developed to explain when and why it works well [11-15]. The other norms also has been used in different papers for identify with convex optimization such as nuclear norms, Tikhonov norm and Chebishov norm. And the other convexity issue has been discussed in other papers [16-18]. But in none of them error and CPU time and order of system that identified didn't checked.

In this paper we will identify a system with a convex optimization method and we will compute

1. Advanced Control Systems Laboratory, School of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, sojoodi@modares.ac.ir
2. Advanced Control Systems Laboratory, School of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

error and order of the answer and CPU-time and compare these values and discover the relationships between them. We will increased data and identify large scale systems with this method and interior point algorithm and compare these methods. And finally we will offer an interior point algorithm with nonlinear objectives.

2. Small and medium sized problems

The simplest norm approximation problem is an unconstrained problem of the form

$$minimize \left\|Ax-b\right\| \tag{1}$$

where $A \in R^{m\times n}$ and $b \in R^{m}$ are problem data, $x \in R^{n}$ is the variable, and $\|.\|$ is a norm on R^{m}. A solution of the norm approximation problem is sometimes called an approximate solution of $Ax \approx b$, in the norm $\|.\|$. The vector $r = Ax - b$ is called the residual for the problem; its components are sometimes called the individual residuals associated with x.

The norm approximation problem is a convex problem, and is solvable, i.e., there is always at least one optimal solution. Its optimal value is zero if and only if $b \in R(A)$.

Inasmuch as our data is pendulous so for smoothing the solution of the simulation we have to use regularization method for cost function and identification.

In this method we need to use prior knowledge. In this paper we assume that our model is an autoregressive moving average model and j is regression vector and we use 2 step-ahead data for smoothing. So we define the function a as below:

$$a = n^{2}\begin{pmatrix} 1 & -2 & 1 & 0 & & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \mathbf{K} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & & 0 & 0 & 0 & 0 \\ & & \mathbf{M} & & \mathbf{O} & & \mathbf{M} & & \\ 0 & 0 & 0 & 0 & & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{L} & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & & 0 & 1 & -2 & 1 \end{pmatrix} \in R^{(n-2)\times n} \tag{2}$$

Now we can define cost function that we will use in our simulation as below:

$$minimize\ \ b\left\|jx-b\right\|_{2}^{2} + l\left\|x\right\|_{2}^{2} + (1-l)\left\|a\right\| \tag{3}$$

Where $b \geq 5max(l)$, and f is regression vector of ARMA model and $0 < l < 1$.

3. Interior point algorithm

For small and medium sized problems the applications discussed in the previous sections can be handled by general-purpose convex optimization solvers, such as the modeling packages CVX [19] and YALMIP [20], and general-purpose conic optimization packages. In this section we discuss algorithmic approaches that are of interest for large problems that fall outside the scope of the general-purpose solvers.

Interior-point algorithms are known to attain a high accuracy in a small number of iterations, fairly independent of problem data and dimensions. The main drawback is the high linear algebra complexity per iteration associated with solving the Newton equations that determine search directions.

However sometimes problem structure can be exploited to devise dedicated interior-point implementations that are significantly more efficient than general purpose solvers. In interior point algorithm we use linear programming, so we start form a description of linear programming.

4. Linear programming

We consider the following primal linear program:

$$\begin{aligned} miiin \quad & c^{T}x \\ s.t. \quad & Ax = b \\ & 0 \leq x_{i} \leq u_{i} \qquad i \in I \\ & 0 \leq x_{i} \qquad i \in J \end{aligned} \tag{4}$$

Where $A \in R^{m\times n}$, which determines the sizes of other vectors involved, and I and J are disjoint index sets such that $I \cup J = \{1,2,...,n\}$ and $I \cap J = \varnothing$.

Without loss of generality, let us assume that for some positive integer $n_u \leq n$

$$I = \{1,2,...,n_u\} \quad and \quad J = \{n_u+1, n_u+1,...,n\}.$$

Given a vector x in R^{n}, we use the notation x_u for the vector in R^{n_u} whose elements are the first n_u elements of, i.e., $[x_u]_i = x_i$, $i = 1,2,...,n_u$.

Moreover, we define the appending operator "app" from R^{n_u} to R^{n} that appends $n-n_u$ zero to vectors in R^{n_u}, i.e., for $w \in R^{n_u}$:

$$[app(w)]_i = \begin{cases} w_i, & 1 \leq i \leq n_u \\ 0, & n_u\ \mathbf{p}\ i \leq n. \end{cases} \tag{5}$$

By adding the slack variable $s \in R^{n_u}$, now we can rewrite the above linear program into the standard form:

$$\begin{aligned} miin \quad & c^{T}x \\ s.t. \quad & Ax = b \\ & x_u + s = u \\ & x \geq 0, s \geq 0 \end{aligned} \tag{6}$$

The dual of the above standard primal linear program is:

$$\begin{aligned} max \quad & b^T y - u^T w \\ s.t. \quad & A^T y + z - app(w) = c \\ & z \geq 0, w \geq 0 \end{aligned} \tag{7}$$

Where y, z *and* w are the dual variables and slack.

It is well known that the solution of the primal and the dual linear programs, if they exist, satisfy the following KKT conditions which is a system of linear quadratic equations with no negativity constraints on some variables:

$$F(x,z,s,w,y) = \begin{pmatrix} Ax - b \\ x_u + s - u \\ A^T y + z - app(w) - c \\ xz \\ sw \end{pmatrix} = 0, \quad (x,z,s,w) \geq 0 \tag{8}$$

where xz and sw denote component wise multiplications and the equations $xz = 0$ and $sw = 0$ are called the complementarily conditions for the linear program.

For nonnegative variables x, s, z, w, we will call the quantity $x^T z + s^T w$ the duality gap. The duality gap measures the residual of the complementarily portion of F in l_1-norm when $(x,z,s,w) \geq 0$. A straightforward calculation shows that the Jacobin matrix of $F(x,z,s,w,y)$ is

$$F'(x,z,s,w,y) = \begin{bmatrix} A & 0 & 0 & 0 & 0 \\ E^T & 0 & I_u & 0 & 0 \\ 0 & I & 0 & -E & A^T \\ Z & X & 0 & 0 & 0 \\ 0 & 0 & W & S & 0 \end{bmatrix}. \tag{9}$$

where $E^T = [I_u \;\; 0]$, I_u is the identity matrix of dimension n_u, $X = diag(x)$, $Z = diag(z)$, $S = diag(s)$ and $W = diag(w)$. Unlike a primal method which concentrates on solving the primal program, a primal interior point method for linear programming solves the KKT, which includes all the primal and dual variables and slacks.

5. Newton's method

Solving a linear program is equivalent to solving a system of linear quadratic equations with no negativity constraints on a subset of variables. In order to simplify our coming discussion, we rewrite the KKT into a constrained algebraic system of l equations and l variables with no negativity constraints on l_+ variables:

$$F(v) = 0, \; v_i \geq 0, \; 1 \leq i \leq l_+ \tag{10}$$

where

$v = (x,z,s,w,y)$, $l = 2n + 2n_u + m$ *and* $l_+ = 2n + 2n_u$.

Let us first drop the no negativity constraints from equation and consider Newton's method for solving an unconstrained system of nonlinear equations $F(v) = 0$.

Which can be written as:

$$v^{k+1} = v^k - F'(v^k)^{-1} F(v^k). \tag{11}$$

It is well known that Newton's method has excellent local convergence properties. More specifically if the Jacobin matrix $F'(v)$ is nonsingular and Lipchitz continuous at a solution v^*, and the initial point v^0 is sufficiently close to v^*, then the iterate sequence $\{v^k\}$ converges to v^* Q-quadratic. On the other hand, Newton's method generally does not have very good global convergence properties. A variant of Newton's method is called the damped Newton:

$$v^{k+1} = v^k - a^k F'(v^k)^{-1} F(v^k) \tag{12}$$

Which introduces a damping factor, or step length, a^k, usually chosen from the interval $(0,1]$, to enhance global convergence. Another variation of Newton's method is the so-called composite Newton's method.

At each iteration, it calculates an intermediate point $\hat{v}^k$ and uses the same Jacobin matrix twice:

$$\hat{v}^k = v^k - F'(v^k)^{-1} F(v^k), \; v^{k+1} = \hat{v}^k - F'(v^k)^{-1} F(\hat{v}^k). \tag{13}$$

Equivalently, we can shorten the expression as

$$v^{k+1} = v^k - F'(v^k)^{-1}(F(v^k) + F(\hat{v}^k)). \tag{14}$$

In terms of linear algebra work, the composite Newton requires one matrix factorization per iteration, same as for Newton's method, but two back-solves instead of one. Since matrix factorization are usually much more expensive than back-solves, the required work per iteration for the composite Newton is comparable with that for Newton's method. However, under similar conditions, the composite Newton has a Q-cubic asymptotic convergence rate, one order faster than that of Newton's method.

Similarly, one can introduce a damping factor into the composite Newton's method:

$$v^{k+1} = v^k - a^k F'(v^k)^{-1}(F(v^k) + F(\hat{v}^k)). \tag{15}$$

6. Simulation results

Examined data in this paper is related to cutaneous potential recordings of a pregnant woman. This data are pendulous and has a large standard deviation, therefore it can't be fitted with ordinary curve fittings, so we use the smoothing sp-line for computing the order of model.

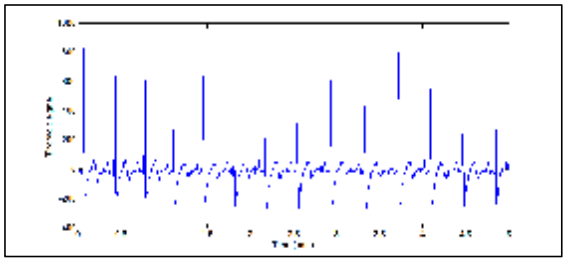

Figure 1. potential recordings of a pregnant woman

Now with assuming cost function that mentioned in previous section we are going to identify the system. For computing the order of the answer we use the smoothing sp-line for fit.

The smoothing spline s is constructed for the specified smoothing parameter p and the specified weights w_i. The smoothing spline minimizes

$$p\sum_i w_i (y_i - s(x_i))^2 + (1-p)\int (\frac{d^2 s}{dx^2})^2 dx \qquad (16)$$

If the weights are not specified, they are assumed to be 1 for all data points. p is defined between 0 and 1.

p = 0 produces a least-squares straight-line fit to the data, while p = 1 produces a cubic sp-line interplant.

Now we can define cost function that we will use in our simulation as below:

$$minimize\ b\left\| j\,x - b\right\|_2^2 + l\left\| x\right\|_2^2 + (1-l)\left\| a\,x\right\|_2^2 \qquad (17)$$

Where $b \geq 5max(l)$, and f is regression vector of ARMA model and $0 < l < 1$.

In curve fitting problems we have a parameter R-square that if this parameter is 1 that means our fit is match with the real data. Here we set $R - square = 0.9$, and compute parameter p for different values for l.

Now if we use interior point method that discussed in previous section we will achieve this information.

Table 1: Model Characteristics for different values for n obtained with interior point algorithm

N	BETA_OPT	E_OPT	P_OPT	CPU_TIME
100	0.95	6.9321	0.984341	336.45
250	0.85	6.4522	0.983903	350.43
500	0.8	5.9555	0.983318	412.74
750	0.65	5.0012	0.983004	503.45
1000	0.6	4.0947	0.98276	602.69
1250	0.5	3.9134	0.98264	650.354
1500	0.35	3.7634	0.98244	714.34
1750	0.3	3.6435	0.98226	769.45
2000	0.2	3.52145	0.98212	823.4

Now if we identify our data with interior algorithm that explained in previous section with nonlinear cost functions and do same things that in previous sections we did, we will achieve these results.

Table 2: Model Characteristics for different values for n obtained with convex optimization, interior point algorithm with and without nonlinear cost functions

N	E(CVX)	CPU(CVX)	E(IPM)	CPU(IPM)	E(IPM. NL)	CPU(IPM. NL)
100	5.0012	0.82	6.9321	336.45	5.4121	388.45
250	4.2053	3.57	6.4522	350.43	4.9322	402.43
500	3.455	12.74	5.9555	412.74	4.4355	464.74
750	2.6031	22.3	5.0012	503.45	3.4812	555.45
1000	1.5946	42.69	4.0947	602.69	2.5747	654.69
1250	1.4032	160.3	3.9134	650.354	2.3934	702.354
1500	1.2043	484.3	3.7634	714.34	2.2434	766.34
1750	1.1432	803.4	3.6435	769.45	2.1235	821.45
2000	1.0218	1523.4	3.52145	823.4	2.00145	875.4

According to Table 2 we see that convex optimization method in small and medium sized problems give us very good CPU time that is less than other methods that offered. And also this method gives us less error than interior point method with and without nonlinear cost functions in these sizes of problems.

But if we use this method in large scale problems, likewise it will give us good error but CPU time increase strongly and with exponential function, that is not desirable, so we cannot use this method in large scale problems. According to the results in Table 2 we see that IPM in large scale has less CPU time in large scale so we should use this method in these types of problems(see figure 3,4). Only thing that remains is the error of these answers that seen is very much than convex problems and maybe is not acceptable. In this section we use interior point method with nonlinear cost function that come in below.

$$minimize\ b\left\| j\,x - b\right\|_2^2 + l\left\| x\right\|_2^2 + g\left\| a\,x\right\|_2^2 + (1-l-g)\left\| f\,x - b\right\|_2^2 \left\| x\right\|_2^2$$

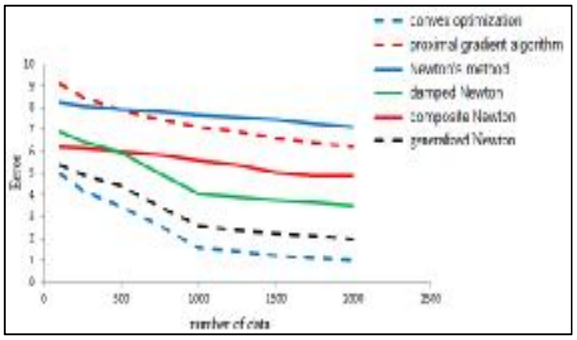

Figure 2. Error of outputs that has been identified with the proposed algorithm

Where $b \geq 5max(l)$, and f is regression vector of ARMA model and $0 < l, g, l + g < 1$.

If we use this cost function and identify our data with interior point method we will achieve the results that come in Table 2.

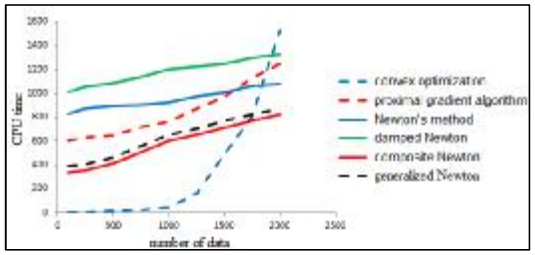

Figure 3. CPU time of identify system with different algorithms

According to the results we see that with this method howbeit CPU time increased a little but we reduce error and this is a good trade-off between error and CPU time. Now we calculate error of outputs that has been identified with the proposed algorithm. In figure 2 we see that the error of our model is soupcon and this model is accepted.

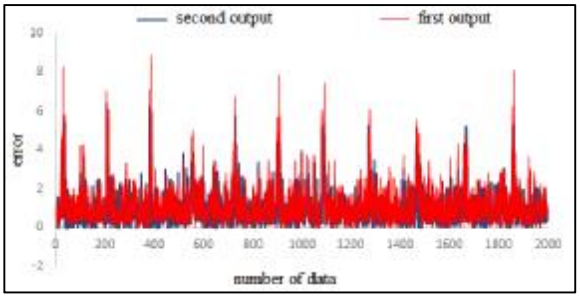

Figure 4. Error of outputs that has been identified with the proposed algorithm

7. Conclusions

Convex optimization method in small and medium sized problems gives us very good CPU time that is less than other methods that offered. And also this method gives us less error than interior point method with and without nonlinear cost functions in these sizes of problems. But if we use this method in large scale problems, likewise it will give us good error but CPU time increase strongly and with exponential function, that is not desirable, so we cannot use this method in large scale problems. We see that IPM in large scale has less CPU time in large scale so we should use this method in these types of problems. With this method howbeit CPU time increased a little but we reduce error and this is a good trade-off between error and CPU time.

References

[1] L. Ljung. System Identification - Theory for the User. Prentice-Hall Upper Saddle River, N.J., 2nd edition, 1999.

[2] R. Pintelon and J. Schoukens. System Identification – A Frequency Domain Approach. IEEE Press, New York, 2nd edition, 2012.

[3] L. Ljung. The System Identification Toolbox: The Manual. The MathWorks Inc. 1st edition 1986, 8th edition 2012, Natick, MA, USA, 2012.

[4] L. Vandenberghe, "Convex optimization techniques in system identification," in Proc. IFAC Symposium on System Identification, 2013, pp. 71–76.

[5] M. Deistler, System identification and time series analysis: Past, present, and future. Springer, 2002.

[6] L. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D, 60:259–268, 1992.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432, 2008.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20:33–61, 1999.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. The Annals of Statistics, 32(2):407– 499, 2004.

[10] E. Cand`es and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. The Annals of Statistics, 35(6):2313–2351, 2007.

[11] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. IEEE Transactions on Information Theory, 47(7):2845–2862, 2001.

[12] E. J. Cand`es and M. B. Wakin. An introduction to compressive sampling. IEEE Signal Processing Magazine, 25(2):21–30, 2008.

[13] D. L. Donoho and J. Tanner. Sparse nonnegative solutions of underdetermined systems by linear programming. Proceedings of the National Academy of Sciences of the United States of America, 102(27):9446–9451, 2005.

[14] D. L. Donoho. Compressed sensing. IEEE Transactions on Information Theory, 52(4):1289–1306, 2006.

[15] E. J. Cand`es, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52(2):489–509, 2006a.

[16] I. R. Manchester, M. M. Tobenkin, and J. Wang, "Identification of nonlinear systems with stable oscillations," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 5792–5797.

[17] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1235–1256, 2009.

[18] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, no. 3, pp. 165–218, 2011.

[19] M. Grant and S. Boyd. *CVX: Matlab software for dis-ciplined convex programming (web page and software)*.http://stanford.edu/~boyd/cvx, 2007.

[20] J. L¨ofberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.