# LINEAR REGRESSION WITH A SPARSE PARAMETER VECTOR

*Erik G. Larsson*

School of EE, Communication Theory
Royal Institute of Technology
Osquldas väg 10, 100 44 Stockholm, Sweden
Email: erik.larsson@ee.kth.se.

*Yngve Selén*

Dept. of Information Technology
Uppsala University
P.O. Box 337, 751 05 Uppsala, Sweden.
Email: yngve.selen@it.uu.se.

**ABSTRACT**

We consider linear regression under a model where the parameter vector is known to be sparse. Using a Bayesian framework, we derive a computationally efficient approximation to the minimum mean-square error (MMSE) estimate of the parameter vector. The performance of the so-obtained estimate is illustrated via numerical examples.

## 1. INTRODUCTION

### 1.1. Problem Formulation

Consider the linear regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{h} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{h}$ is a parameter vector of length $n$, $\boldsymbol{y}$ is an $N$-vector of observations, $\boldsymbol{X}$ is a known $N \times n$ regressor matrix, and $\boldsymbol{e}$ is a vector of noise. The task is to estimate $\boldsymbol{h}$, given that $\boldsymbol{y}$ was observed.

As is well-known, the least-squares (LS) estimate of $\boldsymbol{h}$ is given by the minimizer of $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{h}\|^2$ with respect to $\boldsymbol{h}$, i.e., (see [1], for example):

$$\hat{\boldsymbol{h}}_{\text{LS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{2}$$

If the noise is zero-mean, white and Gaussian, then the LS estimate coincides with the maximum-likelihood (ML) estimate [1]. Hereafter, we shall assume that $\boldsymbol{e} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$.

The LS estimate is commonly used owing to its simplicity and its connection to ML. However, if something is known about $\boldsymbol{h}$ *a priori* (before the data are collected), then one can do better than the LS estimate. For example, if one knows that $\boldsymbol{h} \sim N(\boldsymbol{0}, \gamma^2\boldsymbol{I})$ *a priori*, then the estimate of $\boldsymbol{h}$ which has the smallest mean-square error (MSE, $E[\|\hat{\boldsymbol{h}} - \boldsymbol{h}\|^2]$) is given by the conditional mean of $\boldsymbol{h}$ given that $\boldsymbol{y}$ was observed [1]:

$$\hat{\boldsymbol{h}}_{\text{MMSE}} = E[\boldsymbol{h}|\boldsymbol{y}] = \gamma^2\boldsymbol{X}^T(\gamma^2\boldsymbol{X}\boldsymbol{X}^T + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{3}$$

(Note that when $\gamma^2 \to \infty$, corresponding to the observer having no *a priori* knowledge of $\boldsymbol{h}$, then the MMSE and LS estimates coincide.)

Generally, the minimum MSE (MMSE) estimate is better (in the MSE sense) than the LS estimate,[1] owing to the influence of the *a priori* knowledge of $\boldsymbol{h}$. We now ask the question: If $\boldsymbol{h}$ is known to be *sparse*, that is, some elements of $\boldsymbol{h}$ are likely to be equal to zero, can we do even better than the above MMSE estimate? And if so, how much better can we do?

### 1.2. Related Work on Regression with Sparse Models

Linear regression for sparse models has been studied before, both in the statistics community and in the signal processing literature. The main approaches that we are aware of include the following:

**1.** The Lasso method [2] estimates $\boldsymbol{h}$ by minimizing the LS criterion $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{h}\|^2$ subject to a $L_1$-norm constraint on the parameter vector. More specifically, Lasso finds $\boldsymbol{h}$ via:[2]

$$\hat{\boldsymbol{h}}_{\text{Lasso}} = \arg\min_{\boldsymbol{h}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{h}\|^2 \qquad \text{subject to } \sum_{j=0}^{n-1} |h_j| \leq c. \tag{4}$$

The parameter $c$ is a user parameter. Although it is perhaps not immediately apparent, using a small enough value for $c$ typically leads to a sparse parameter vector estimate. That is, the estimated $\boldsymbol{h}$ will have many of its coefficients equal to zero. Lasso requires that the parameter $c$ is chosen by the user, or "estimated" from the data in some way. This is a non-trivial task which is discussed in [2].

There also exist other more recent methods related to Lasso, such as Forward stagewise regression and Lars (Least angle regression) [3]. They also require the choice of a user parameter like $c$.

**2.** Another approach [4] (see also [5]) is to compute a Bayesian estimate of $\boldsymbol{h}$ assuming an *a priori* density on $\boldsymbol{h}$ which encourages sparseness. Typically, this prior distribution has a sharp peak at zero. A specific example of such a prior is

$$p(\boldsymbol{h}) \sim \exp\left(-\sum_{j=0}^{n-1} |h_j|^p\right) \tag{5}$$

where $p$, $0 < p \leq 1$, is a user parameter to be chosen.

**3.** In [6] the authors suggested to estimate $\boldsymbol{h}$ via[3]

$$\hat{\boldsymbol{h}}_{p\text{-norm}} = \arg\min_{\boldsymbol{h}} \left(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{h}\|_p + \lambda\|\boldsymbol{h}\|_p\right) \tag{6}$$

where $\lambda$ and $p$ are user parameters. (We note the similarity to (4). Like for Lasso, the criterion (6) effectively favors sparse parameter

This work was supported in part by the Swedish Science Council (VR). The work was initiated when the first author was with the George Washington University, Washington DC, USA, and supported in part by the National Science Foundation grant CCF-0429228.

[1] Unlike the LS estimate, however, the MMSE estimate is biased.

[2] Notation: here $h_j$ is the $j$th element of $\boldsymbol{h}$.

[3] Here, $\|\cdot\|_p$ stands for the $L_p$-norm.

vectors $\boldsymbol{h}$. However, Lasso is not a special case of (6).) The user parameter $\lambda$ balances the conflicting objectives of minimizing the residual (this requires a small value of $\lambda$) and obtaining an estimate with a sparse structure (this calls for a large $\lambda$).

The above methods allow (or in some applications even assume, see [7] for a suggestion similar to (6)) the linear regression problem to be overcomplete, that is $n > N$.

We note that none of the methods above has any clear connections to the sparseness structure of the model in terms of the probability of a given coefficient being equal to zero. The goal of this paper is to present an estimator for which this connection is explicit.

### 1.3. Contribution of This Work

We propose a method for computing the MMSE parameter vector estimate under the explicit *a priori* assumption that a given coefficient of $\boldsymbol{h}$ is equal to zero with a certain probability. The method is computationally very efficient (typically, a $50 \times 10$ regression model takes about 15 ms on a standard desktop PC). Our method is Bayesian, and as such it requires certain *a priori* assumptions. Specifically, in addition to the variances $\sigma^2$ and $\gamma^2$, the algorithm takes as input the probability $p$, which describes how likely it is (before the data are observed) that any given coefficient of $\boldsymbol{h}$ is equal to zero. The *a priori* parameters required by the algorithm have clear and unambiguous interpretations, and are explicit in the estimator and its derivation. Also, by varying these parameters one can directly study how the estimates are affected. It turns out that the estimator is relatively insensitive to the choices of $p, \sigma^2, \gamma^2$. See Sections 2.3 and 3 for a further discussion of the role of the *a priori* parameters.

## 2. THE MMSE ESTIMATE UNDER A SPARSENESS CONSTRAINT

### 2.1. Model

We shall assume that $\boldsymbol{h}$ has a sparse structure which can be described as a mixture of $2^n$ components, $H_0, ..., H_{2^n-1}$. That is,

$$p(\boldsymbol{h}) = \sum_{i=0}^{2^n-1} p(\boldsymbol{h}|H_i)P(H_i).$$

For each mixture component $H_i$, a subset of the coefficients of $\boldsymbol{h}$ is constrained to zero and the other coefficients are i.i.d. random variables. Specifically,

$$\begin{cases} H_0: & h_0, ..., h_{n-1} \text{ i.i.d. } N(0,\gamma^2) \\ H_1: & h_0 = 0; h_1, ..., h_{n-1} \text{ i.i.d. } N(0,\gamma^2) \\ H_2: & h_1 = 0; h_0, h_2, ..., h_{n-1} \text{ i.i.d. } N(0,\gamma^2) \\ \vdots & \\ H_n: & h_{n-1} = 0; h_0, ..., h_{n-2} \text{ i.i.d. } N(0,\gamma^2) \\ H_{n+1}: & h_0 = h_1 = 0; h_2, ..., h_{n-1} \text{ i.i.d. } N(0,\gamma^2) \\ H_{n+2}: & h_0 = h_2 = 0; h_1, h_3, ..., h_{n-1} \text{ i.i.d. } N(0,\gamma^2) \\ \vdots & \\ H_{2^n-1}: & h_0 = ... = h_{n-1} = 0. \end{cases} \quad (7)$$

Each mixture component has an associated probability $P(H_i)$. Naturally, these probabilities sum to one:

$$\sum_{i=0}^{2^n-1} P(H_i) = 1.$$

If the coefficients of $\boldsymbol{h}$ are independent and equal to zero with probability $p$ (an assumption which is common to make in practice, but which is *not* necessary for the analysis to come), then we have

$$\begin{aligned} P(H_0) &= (1-p)^n \\ P(H_1) &= p(1-p)^{n-1} \\ P(H_{n+1}) &= p^2(1-p)^{n-2} \\ &\vdots \\ P(H_{2^{n-1}}) &= p^n. \end{aligned}$$

The probabilities $\{P(H_i)\}$, as well as $\{\gamma^2, \sigma^2\}$, are assumed to be known, or at least they are set to something sensible. Typically, like in all Bayesian inference, the priors are set to the best "belief" one has before the data are observed (see Section 2.3).

### 2.2. Computing the MMSE Estimate

The task we want to tackle is that of computing the MMSE estimate of $\boldsymbol{h}$, given $\boldsymbol{y}$, under the mixture model (7). This is equal to the conditional mean, $\hat{\boldsymbol{h}}_{\text{MMSE}} = E[\boldsymbol{h}|\boldsymbol{y}]$. We have

$$\hat{\boldsymbol{h}}_{\text{MMSE}} = E[\boldsymbol{h}|\boldsymbol{y}] = \sum_{i=0}^{2^n-1} P(H_i|\boldsymbol{y})E[\boldsymbol{h}|\boldsymbol{y}, H_i]. \quad (8)$$

In principle, the sum in (8) can be computed (assuming we can calculate its terms). The difficulty is that for large $n$, the number of terms can be unbearably large (as they grow exponentially with $n$), and the computational complexity may become unreasonable.

In order to approximate (8) to obtain a computationally feasible expression, we note that the terms $E[\boldsymbol{h}|\boldsymbol{y}, H_i]$ should be of the same order of magnitude, at least for the values of $i$ for which $P(H_i|\boldsymbol{y})$ is significantly different from zero. Therefore, the weighted sum is dominated by the terms for which $P(H_i|\boldsymbol{y})$ are large, and a good approximation to the MMSE estimate can be obtained by truncating the sum in (8).

Let $\Omega$ be the set of indices $i$ for which $P(H_i|\boldsymbol{y})$ are significant, and then normalize $P(H_i|\boldsymbol{y})$ for $i \in \Omega$ so that they add up to one. Then we arrive at the following approximation to (8),

$$\hat{\boldsymbol{h}}_{\text{MMSE}} \approx \hat{\tilde{\boldsymbol{h}}}_{\text{MMSE}} \triangleq \frac{1}{\sum_{j\in\Omega} P(H_j|\boldsymbol{y})} \sum_{i\in\Omega} P(H_i|\boldsymbol{y})E[\boldsymbol{h}|\boldsymbol{y}, H_i]. \quad (9)$$

By Bayes' rule, we have

$$\hat{\tilde{\boldsymbol{h}}}_{\text{MMSE}} \triangleq \frac{1}{\sum_{j\in\Omega} p(\boldsymbol{y}|H_j)P(H_j)} \sum_{i\in\Omega} P(H_i)p(\boldsymbol{y}|H_i)E[\boldsymbol{h}|\boldsymbol{y}, H_i]. \quad (10)$$

In order to evaluate (10), what remains is the following: (1) To compute $p(\boldsymbol{y}|H_i)$, (2) to compute $E[\boldsymbol{h}|\boldsymbol{y}, H_i]$, and (3) to invent a mechanism for selecting $\Omega$. Before we proceed with these tasks, let us introduce the following notation:

$$\begin{cases} \boldsymbol{x}_j: & j\text{th column of } \boldsymbol{X} \\ \Gamma_i: & \text{the set of indices } j \text{ for which } h_j \text{ are constrained to zero under } H_i \\ \bar{\boldsymbol{h}}_i: & \boldsymbol{h} \text{ with the elements corresponding to } \Gamma_i \text{ removed} \\ \bar{\boldsymbol{X}}_i: & \boldsymbol{X} \text{ with the columns with indices in } \Gamma_i \text{ removed} \\ \tilde{\boldsymbol{h}}_i: & \boldsymbol{h} \text{ with the elements not in } \Gamma_i \text{ removed (i.e., the opposite of } \bar{\boldsymbol{h}}_i). \end{cases}$$

*2.2.1. Computation of $p(\boldsymbol{y}|H_i)$*

Conditioned on $H_i$, we have that $h_j = 0$ for $j \in \Gamma_i$ and $h_j$ i.i.d. $N(0, \gamma^2)$ for $j \notin \Gamma_i$. Then,

$$\boldsymbol{y}|H_i \sim N(\boldsymbol{0}, \boldsymbol{Q}_i)$$

where

$$\boldsymbol{Q}_i = \gamma^2 \sum_{j \notin \Gamma_i} \boldsymbol{x}_j \boldsymbol{x}_j^T + \sigma^2 \boldsymbol{I} = \gamma^2 \bar{\boldsymbol{X}}_i \bar{\boldsymbol{X}}_i^T + \sigma^2 \boldsymbol{I}.$$

So,

$$p(\boldsymbol{y}|H_i) = \frac{1}{\sqrt{2\pi}^N} \frac{1}{|\boldsymbol{Q}_i|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{y}^T \boldsymbol{Q}_i^{-1} \boldsymbol{y}\right). \quad (11)$$

*2.2.2. Computation of $E[\boldsymbol{h}|\boldsymbol{y}, H_i]$*

It is sufficient to find $E[\tilde{\boldsymbol{h}}_i|\boldsymbol{y}, H_i]$ and $E[\bar{\boldsymbol{h}}_i|\boldsymbol{y}, H_i]$, because $\boldsymbol{h}$ is composed of $\tilde{\boldsymbol{h}}, \bar{\boldsymbol{h}}$. Clearly

$$E[\tilde{\boldsymbol{h}}_i|\boldsymbol{y}, H_i] = \boldsymbol{0} \quad (12)$$

since the elements of $\tilde{\boldsymbol{h}}_i$ are constrained to zero under $H_i$. We must now find $E[\bar{\boldsymbol{h}}_i|\boldsymbol{y}, H_i]$.

Under $H_i$, $\bar{\boldsymbol{h}}_i$ and $\boldsymbol{y}$ are jointly Gaussian as follows

$$\begin{bmatrix} \boldsymbol{y} \\ \bar{\boldsymbol{h}}_i \end{bmatrix} |H_i \sim N\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{Q}_i & \gamma^2 \bar{\boldsymbol{X}}_i \\ \gamma^2 \bar{\boldsymbol{X}}_i^T & \gamma^2 \boldsymbol{I} \end{bmatrix}\right).$$

Applying a standard result (Theorem 10.2 of [1], for example), the conditional mean evaluates to

$$E[\bar{\boldsymbol{h}}_i|\boldsymbol{y}, H_i] \quad = \quad \gamma^2 \bar{\boldsymbol{X}}_i^T \boldsymbol{Q}_i^{-1} \boldsymbol{y}. \quad (13)$$

*2.2.3. Selection of $\Omega$*

We now have the ingredients of the sum in (10); namely, equations (11), (12) and (13). What remains is to select the subset of $H_i$, over which the sum in (10) should be computed. We propose a strategy, based on successive model reduction, which results in the following algorithm:

1. Start with $i = 0$; i.e., consider $H_0$.
2. Compute the contribution of $H_i$ to (10).
3. Find out what coefficient $h_j$ would reduce $P(H_i|\boldsymbol{y})$ by the least amount, if it were constrained to zero. That is, evaluate $P(H_k|\boldsymbol{y})$ for all $H_k$ which can be obtained from $H_i$ by constraining one more coefficient to zero. Then eliminate this coefficient from consideration; i.e., let $i := k$.
4. If $i = 2^n - 1$ (this must happen after $n$ iterations), then compute the contribution of $H_i$ to (10) and terminate. Otherwise, go to 2.

### 2.3. Discussion of the *a Priori* Assumptions

We have assumed that $p$, $\gamma^2$, $\sigma^2$ are known. In principle, nothing prevents the user from "estimating" these parameters from the data. Doing so would give an "empirical Bayesian" estimate [8]. However, we refrain from this as we believe that "automatic" procedures for choosing user parameters are often influenced by hidden assumptions. Furthermore, if one accepts the Bayesian philosophy one can argue that by definition, $p$, $\gamma^2$, $\sigma^2$ are *a priori* parameters and should therefore not depend on the observations: estimating $p$, $\gamma^2$, $\sigma^2$ from the data would void the optimality (in the Bayesian MMSE sense) of the method. One could, however, eliminate the explicit dependence of the estimates on $p$, $\gamma^2$, $\sigma^2$ within the Bayesian framework by treating these as random variables with certain (non-informative) *a priori* distributions. Doing so would likely lead to an estimator which is computationally difficult to compute (possibly Monte Carlo methods [9] could be used).

## 3. NUMERICAL EXAMPLES

We consider a relatively well-conditioned regressor matrix. Specifically, we let the elements in $\boldsymbol{X}$ be i.i.d. samples from a $N(0, 1)$ distribution. $\boldsymbol{X}$ is set to be of dimensions $50 \times 10$ (i.e., $N = 50$, $n = 10$). We compare the new sparse MMSE estimator (9) to the conventional LS estimate (2) and the MMSE estimate (3). Unless specified otherwise, we supply the estimators with the true values of $p$, $\gamma^2$ and $\sigma^2$, set to $p_0 = 0.5$, $\gamma_0^2 = 1$ and $\sigma_0^2 = -15$ dB, respectively.

We also compare with the Lasso method [2]. For this comparison, we used the official implementation of Lasso [10]. The routines in [10] were used to automatically choose $c$.

The methods are evaluated via Monte-Carlo simulations. As performance measure we use the empirical MSE of the parameter estimates, MSE $= M^{-1} \sum_{m=1}^{M} \|\hat{\boldsymbol{h}}^{(m)} - \boldsymbol{h}^{(m)}\|^2$, where $\hat{\boldsymbol{h}}^{(m)}$ and $\boldsymbol{h}^{(m)}$ denote the estimated and true parameter values for realization number $m$. $M = 10000$ is our total number of Monte-Carlo runs.

First, we generated sparse data, using the default parameter values above, but varying the noise variance $\sigma_0^2$. In this example our estimator is perfectly matched to the data model. Figure 1 shows the results. Note that the estimator significantly outperforms the Lasso, LS and the non-sparse MMSE methods.

Next, we study the effect of using an estimator mismatched to the data. First we vary the sparseness $p_0$ and supply the estimator with mismatched $p$-values. The results are shown in Figure 2. To obtain Figure 3 we vary $\gamma_0^2$ and use the sparse MMSE estimator with mismatched $\gamma^2$-values. In Figure 4 we vary $\sigma_0^2$ and try estimators with mismatched values of $\sigma^2$. We show the performances of the LS and the Lasso estimators together with three sparse MMSE estimators with different values of the studied *a priori* parameter.

We see that the proposed estimator is very robust to mismatched parameter values. Indeed, in Figure 2, the sparse MMSE estimator outperforms the other estimators except when $p_0 < 0.1$. Figure 3 shows that the estimator is very insensitive to the choice of $\gamma^2$; even for severely mismatched $\gamma^2$-values the estimator still outperforms the LS and the Lasso estimators. From Figure 4 we again note the estimator's robustness against mismatched parameters. If the error in $\sigma^2$ is lower than 5 dB, the LS and Lasso methods are outperformed by the sparse MMSE estimator.

## 4. CONCLUDING REMARKS

Sparse models have diverse applications. For example, in a statistical data analysis one may know before the measurement that the data are likely to be explained by only a few factors. Another instance where sparse linear models are relevant is the estimation of communication channel impulse responses [11]. Yet another application is in wavelet theory [12].

We derived an approximation to the MMSE estimate for a linear regression model under a sparseness assumption. The estimate shows good performance, it is computationally efficient and it is very

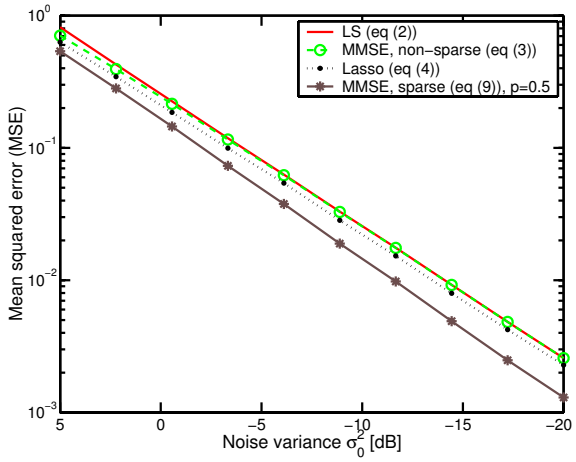


**Fig. 1**. Performance example: Matched estimator, $p_0 = p = 0.5$.

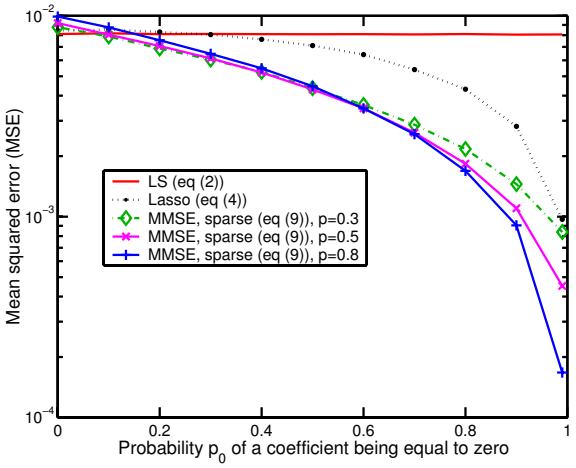


**Fig. 2**. Performance example: $p$-mismatch.

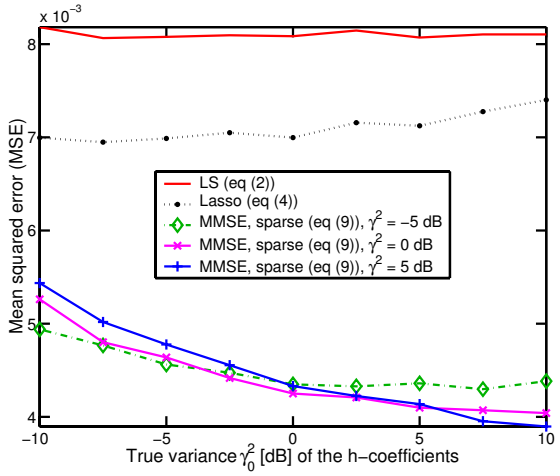


**Fig. 3**. Performance example: $\gamma^2$-mismatch.

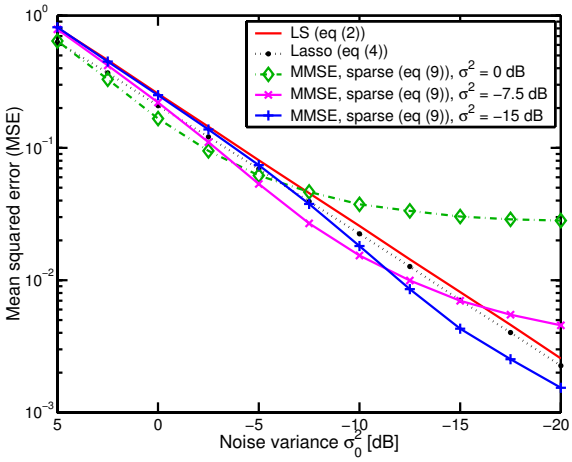


**Fig. 4**. Performance example: $\sigma^2$-mismatch.

robust to mismatches in the *a priori* assumptions. An implementation of the estimator (in C++) can be obtained from [13].

## 5. REFERENCES


[1] S. Kay, "Fundamentals of statistical signal processing: Estimation theory", Prentice-Hall, 1993.

[2] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[3] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, pp. 407–499, April 2004.

[4] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Transactions on Signal Processing*, vol. 51, pp. 760–770, March 2003.

[5] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2153–2164, August 2004.

[6] M. S. O'Brien, A. N. Sinclair and S. M. Kramer, "Recovery of a sparse spike time series by $L_1$ norm deconvolution," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3353–3365, December 1994.

[7] J.-J. Fuchs, "On sparse representation in arbitrary redundant bases," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1341–1344, June 2004.

[8] E. I. George and D. P. Foster, "Calibration and empirical Bayes variable selection," *Biometrika*, vol. 87, no. 4, pp. 731–747, December 2000.

[9] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[10] `http://www-stat.stanford.edu/~hastie/Papers/LARS/`

[11] J. Homer, I. Mareels, R. R. Bitmead, B. Wahlberg and F. Gustafsson, "LMS Estimation via Structural Detection," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2651–2663, October 1998.

[12] I. M. Johnstone and B. W. Silverman, "EbayesThresh: R Programs for Empirical Bayes Thresholding," *Journal of Statistical Software*, vol. 12, Apr. 2005.

[13] `http://www.s3.kth.se/~elarsso/software/`