NTIA Technical Memorandum TM-03-397

An Iterated Nested Least-Squares Algorithm for Fitting Multiple Data Sets

Stephen D. Voran

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary

Nancy J. Victory, Assistant Secretary
for Communications and Information

October 2002

ACKNOWLEDGEMENT

The author gratefully acknowledges the contributions of Stephen Wolf to this work. He was the first to apply the INLS and AINLS algorithms to objective and subjective video quality data and provided valuable feedback on the algorithms.

CONTENTS

Page

ABSTRACT 1

1. INTRODUCTION 1

2. ALGORITHM DEVELOPMENT 4

   2.1 Preliminaries 4

   2.2 Conventional Least-Squares is Not Sufficient 5

   2.3 Apportioning the Fitting Error 6

   2.4 Removing Excess Degrees of Freedom 7

   2.5 The Iterated Nested Least-Squares (INLS) Algorithm 8

3. EXAMPLE ALGORITHM RESULTS 11

   3.1 Simulated Data Results 11

   3.2 Actual Data Results 14

4. SUMMARY 18

5. REFERENCES 18

AN ITERATED NESTED LEAST-SQUARES ALGORITHM FOR FITTING MULTIPLE DATA SETS

Stephen D. Voran[1]

A multiple data set fitting problem often arises in conjunction with the development of objective estimators of perceived audio or video quality. In such development work, we often seek the best linear relationship between a set of objective audio or video quality estimation parameters and a set of subjective audio or video quality scores. In order to find the most robust and reliable relationship, we prefer to perform a least-squares fit using as many audio or video data points as possible. This motivates us to combine scores from different subjective tests. Unfortunately, scores from different subjective tests or data sets can differ in significant ways due to differing test procedures, environments, languages, and other sources. We develop a solution to this multiple data set fitting problem: the iterated nested least-squares (INLS) algorithm. This algorithm iterates between two least-squares steps. One step attempts to homogenize heterogeneous data sets through the use of a single first-order correction for all of the data points in each data set. The other least-squares step solves for the appropriate linear combination of the parameters, across all data sets. We also offer example INLS algorithm results using simulation data and data from telephone-bandwidth speech quality tests. For convenience we have written this memorandum in the language of objective estimation of perceived audio and video quality but the results are completely general and can be used to fit other types of data sets as well.

Key words: audio quality estimation, data set fitting, least-squares fitting, linear regression, meta-analysis, speech quality estimation, video quality estimation

1. INTRODUCTION

Least-squares fits are often used to relate various types of experimental data. A very common situation involves relating a set of independent variables (or experiment control parameters) to a dependent variable (or experiment outcome). More generally though, one can use least-squares fitting to find relationships between multiple measurements of events. An interesting and unique multiple measurements problem (or multiple data set fitting problem) often arises in conjunction with the development of objective estimators of perceived audio or video quality. Examples of

[1] The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80305.

recent work in these areas can be found in [1]-[10]. In this type of development work, we often seek the best linear relationship between a set of r objective audio or video quality estimation parameters (parameters) and a set of subjective audio or video quality scores (scores). The parameters are objectively computed from audio or video signals with the goal of quantifying the perceptually relevant components of audio or video signal distortion. The scores are gathered through subjective tests where human subjects hear or see audio or video signals and then provide their opinions of audio or video quality on some scale. See [11] for examples of subjective test procedures and scales for telephone-bandwidth speech signals.

When r parameters are available for n audio or video data points, we form r column vectors $\boldsymbol{p}_i$, i=1 to r, and each column vector has length n. Thus $\boldsymbol{p}_i$ contains the values of the i^{th} parameter for the n data points. We then build the n by r parameter matrix

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_r]. \tag{1}$$

We also arrange the n corresponding scores into the length n column vector $\boldsymbol{s}$. We can then solve the least-squares problem

$$\boldsymbol{s} \approx \hat{\boldsymbol{s}} = \boldsymbol{Pw} \tag{2}$$

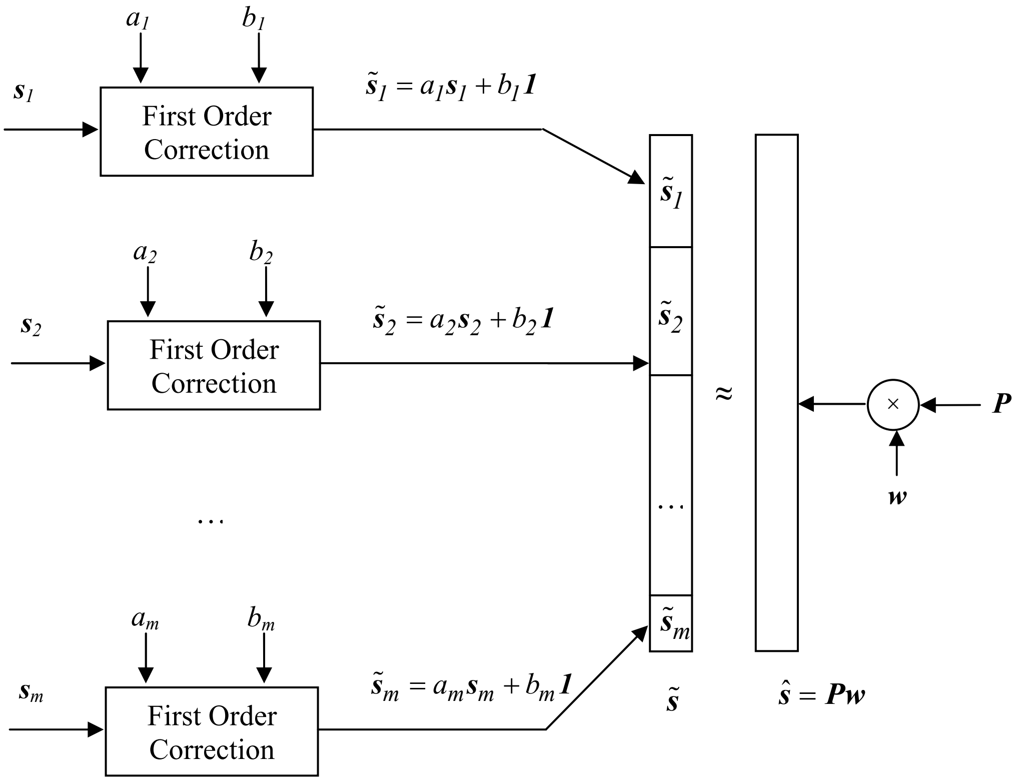
to find the set of weights $\boldsymbol{w}$ that describe the linear relationship between the parameters and score. Once this relationship has been determined to our satisfaction, we can use objectively computed parameters and $\boldsymbol{w}$ to estimate the subjective scores in situations where no subjective scores are available.

In order to find the most robust and reliable relationship, we would like the least-squares fit (2) to use as many audio or video data points as possible (i.e., we would like to maximize n). This motivates us to combine scores from different subjective tests. Unfortunately, the scores from different subjective tests can differ in ways that may not be known. These differences may stem from differing test procedures, differing test environments, cultural and language differences, and even different test scales. In an attempt to homogenize these heterogeneous scores and bring them all to a common scale, we may elect to allow a single first-order correction for all of the scores in each subjective test,

$$\tilde{\boldsymbol{s}} = a\boldsymbol{s} + b\boldsymbol{1}, \tag{3}$$

where $\boldsymbol{s}$ and $\tilde{\boldsymbol{s}}$ are the original and corrected score vectors and $\boldsymbol{1}$ is a column vector of ones. We refer to the scores and corresponding parameters from m different subjective tests as different data sets. Thus we describe the process of combining scores from different tests as a multiple data set fitting problem. This multiple data set fitting scenario is fully described in

Figure 1. Given the scenario described in Figure 1 we must identify appropriate values for $\{a_i\}_{i=1}^m$, $\{b_i\}_{i=1}^m$, and $\boldsymbol{w}$. Solving for these quantities is the topic of this memorandum.



a_1
b_1
$\boldsymbol{s}_1$
First Order Correction
$\tilde{\boldsymbol{s}}_1 = a_1\boldsymbol{s}_1 + b_1\boldsymbol{1}$
a_2
b_2
$\boldsymbol{s}_2$
First Order Correction
$\tilde{\boldsymbol{s}}_2 = a_2\boldsymbol{s}_2 + b_2\boldsymbol{1}$
…
a_m
b_m
$\boldsymbol{s}_m$
First Order Correction
$\tilde{\boldsymbol{s}}_m = a_m\boldsymbol{s}_m + b_m\boldsymbol{1}$
$\tilde{\boldsymbol{s}}_1$
$\tilde{\boldsymbol{s}}_2$
…
$\tilde{\boldsymbol{s}}_m$
$\tilde{\boldsymbol{s}}$
$\approx$
$\hat{\boldsymbol{s}} = \boldsymbol{Pw}$
×
$\boldsymbol{P}$
$\boldsymbol{w}$

Figure 1. Block diagram showing multiple data set fitting scenario.

In the following we develop our solution to this multiple data set fitting problem: the iterated nested least-squares (INLS) algorithm. We provide a full algorithmic description, and we offer example simulation data and example data from telephone-bandwidth speech quality tests. We also describe a variant algorithm, the anchored iterated nested least-squares (AINLS) algorithm. Throughout this memorandum we use the language of objective estimation of perceived audio and video quality, often referring to parameters and scores. We use this language because it seems more readable than a more generic terminology, but we note that the results are completely general and can be applied to any type of data that requires fitting according to Figure 1. We also note that when one uses the procedure of Figure 1 to relate multiple realizations of a single experiment, the result might be considered to be a simple form of meta-analysis [12].

2. ALGORITHM DEVELOPMENT

2.1 Preliminaries

We are seeking a linear relationship between a set of *r* parameters and a subjective score. We have *n* data points and would like to use all of them in a least-squares fit between the scores and the parameters. The scores come from *m* different data sets and the i^{th} data set has n_i data points:

$$n = \sum_{i=1}^{m} n_i \ . \tag{4}$$

The scores from the *m* data sets are not homogeneous. To homogenize the scores, we allow a single first-order correction for all of the scores in each data set:

$$\tilde{\boldsymbol{s}}_i = a_i \boldsymbol{s}_i + b_i \boldsymbol{1}, \quad i = 1 \, \text{to} \, m, \tag{5}$$

where $\boldsymbol{1}$ is a column vector of n_i ones.

We combine the corrected scores into a single *n* by 1 vector $\tilde{\boldsymbol{s}}$ for notational convenience:

$$\tilde{\boldsymbol{s}}^{\mathrm{T}} = \left[\tilde{\boldsymbol{s}}_1{}^{\mathrm{T}}, \tilde{\boldsymbol{s}}_2{}^{\mathrm{T}}, \tilde{\boldsymbol{s}}_3{}^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{s}}_m{}^{\mathrm{T}} \right] . \tag{6}$$

We also have *n* values from each of *r* parameters. These *n* values correspond to the *n* scores. We form the *n* by *r*+1 parameter matrix ***P***, where columns 1 through *r* contain parameters, and column *r*+1 holds the constant value 1:

$$\boldsymbol{P} = \left[\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r, \boldsymbol{1} \right] . \tag{7}$$

The *r*+1 by 1 weight vector ***w*** allows us to form a linear combination of the columns of ***P*** with the goal of approximating $\tilde{\boldsymbol{s}}$:

$$\tilde{\boldsymbol{s}} \approx \hat{\boldsymbol{s}} = \boldsymbol{P}\boldsymbol{w}, \tag{8}$$

where

$$\hat{\boldsymbol{s}}^{\mathrm{T}} = \left[\hat{\boldsymbol{s}}_1{}^{\mathrm{T}}, \hat{\boldsymbol{s}}_2{}^{\mathrm{T}}, \hat{\boldsymbol{s}}_3{}^{\mathrm{T}}, \ldots, \hat{\boldsymbol{s}}_m{}^{\mathrm{T}}\right]. \tag{9}$$

Finally, we introduce the n by n diagonal cost matrix $\boldsymbol{C}$. $\boldsymbol{C}$ contains entries $0 < c_j$, $j = 1$ to n. This cost matrix allows us to shape the fitting errors. For example, we can force smaller errors where we have higher confidence in the data, if we are willing to accept larger errors where we have less confidence in the data. The next three subsections provide other necessary background development that leads to the INLS algorithm.

2.2 Conventional Least-Squares is Not Sufficient

If $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ were known, we could calculate $\tilde{\boldsymbol{s}}$ using (5) and (6) and then find the cost-weighted least squares solution to (8):

$$\min_{\boldsymbol{w}} \left|\boldsymbol{C}\left(\tilde{\boldsymbol{s}} - \hat{\boldsymbol{s}}\right)\right|^2 \quad \Rightarrow \quad \boldsymbol{w} = \left(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{P}\right)^{-1}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{C}^2\tilde{\boldsymbol{s}}\,. \tag{10}$$

On the other hand, if $\boldsymbol{w}$ were known, we could calculate $\hat{\boldsymbol{s}}$ using (8) and then solve the m cost-weighted least-squares problems for $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$. Let

$$\boldsymbol{S}_i = \left[\boldsymbol{s}_i, \boldsymbol{1}\right], \tag{11}$$

then the m cost-weighted least-squares problems and solutions are

$$\min_{a_i, b_i} \left|\boldsymbol{C}\left(\hat{\boldsymbol{s}}_i - \boldsymbol{S}_i\begin{bmatrix} a_i \\ b_i \end{bmatrix}\right)\right|^2 \quad \Rightarrow \quad \begin{bmatrix} a_i \\ b_i \end{bmatrix} = \left(\boldsymbol{S}_i{}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{S}_i\right)^{-1}\boldsymbol{S}_i{}^{\mathrm{T}}\boldsymbol{C}^2\hat{\boldsymbol{s}}_i\,, \quad i = 1 \text{ to } m. \tag{12}$$

In practice, neither $\boldsymbol{w}$ nor $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ are known *a priori*. Further, the two least squares problems do not combine into a single least-squares problem. One way to find a solution for $\boldsymbol{w}, \{a_i\}_{i=1}^m$, and $\{b_i\}_{i=1}^m$ is to iterate between (10) and (12). This is the basis for the iterated nested least-squares algorithms.

2.3 Apportioning the Fitting Error

We can think of both the scores $\boldsymbol{s}$ and the combined parameters $\hat{\boldsymbol{s}}$ as noisy or errored measurements of some true (but unknowable) underlying global mean perceived audio or video quality value. The scores have errors in them because they are estimated means based on a finite (and often rather small) sample. The combined parameters have errors because they are only estimators of the true underlying perceived audio or video quality. We would like our multiple data set fitting algorithm to acknowledge that both $\boldsymbol{s}$ and $\hat{\boldsymbol{s}}$ are errored. In other words, we would like to apportion the least-squares residual to both sources rather than to just one source or the other. Toward that end we define parameter error vectors $\left\{\boldsymbol{\varepsilon}_{\boldsymbol{p}_i}\right\}_{i=1}^{m}$ and score error vectors $\left\{\boldsymbol{\varepsilon}_{\boldsymbol{s}_i}\right\}_{i=1}^{m}$:

$$a_i\left(\boldsymbol{s}_i+\varepsilon_{s_i}\right)+b_i\boldsymbol{1}=\hat{\boldsymbol{s}}_i+\varepsilon_{p_i},\quad i=1\text{ to }m. \tag{13}$$

In [13] we develop several techniques for finding $\{a_i\}_{i=1}^{m}$ and $\{b_i\}_{i=1}^{m}$ while constraining the cost-weighted error power ratio

$$\frac{\left|\boldsymbol{C}_i\boldsymbol{\varepsilon}_{\boldsymbol{s}_i}\right|^2}{\left|\boldsymbol{C}_i\boldsymbol{\varepsilon}_{\boldsymbol{p}_i}\right|^2}=r_i^2,\quad i=1\,\text{to}\,m. \tag{14}$$

In (14) $\boldsymbol{C}_i$ refers to the n_i by n_i diagonal cost matrix taken from the appropriate portion of the n by n cost matrix $\boldsymbol{C}$. Further, $\boldsymbol{C}_i$ must be scaled so that its squared diagonal entries sum to one:

$$\sum_{j=1}^{n_i}c_{jj}^2=1. \tag{15}$$

The cost-weighted error power ratio $0<r_i^2$ allows us to distribute the total fitting error between the score error $\boldsymbol{\varepsilon}_{\boldsymbol{s}_i}$ and the parameter error $\boldsymbol{\varepsilon}_{\boldsymbol{p}_i}$ in accordance with any prior knowledge we might have of the two error processes.

In this memorandum we use the Direct Estimation (DE) Algorithm to find $\{a_i\}_{i=1}^{m}$ and $\{b_i\}_{i=1}^{m}$ subject to the constraints of (14). The algorithm is fully described in [13]. For the i^{th} data set the algorithm yields

$$a_i = \begin{cases} \dfrac{\left(\dfrac{|\hat{\boldsymbol{y}}|}{|\hat{\boldsymbol{x}}|} r_i^2 - \dfrac{|\hat{\boldsymbol{x}}|}{|\hat{\boldsymbol{y}}|}\right) + \sqrt{\left(\dfrac{|\hat{\boldsymbol{y}}|}{|\hat{\boldsymbol{x}}|} r_i^2 - \dfrac{|\hat{\boldsymbol{x}}|}{|\hat{\boldsymbol{y}}|}\right)^2 + 4 r_i^2 \rho^2}}{2 r_i^2 \rho}, & \rho \neq 0, \\ 0, & \rho = 0, \end{cases} \tag{16}$$

and

$$b_i = m_y - a_i m_x, \tag{17}$$

where

$$\begin{aligned} &\hat{\boldsymbol{x}} = \boldsymbol{C}_i \left(\boldsymbol{s}_i - m_x \mathbf{1} \right), \quad \hat{\boldsymbol{y}} = \boldsymbol{C}_i \left(\hat{\boldsymbol{s}}_i - m_y \mathbf{1} \right), \\ &m_x = \sum_{j=1}^{n_i} c_{jj}^2 s_j, \qquad m_y = \sum_{j=1}^{n_i} c_{jj}^2 \hat{s}_j \quad , \quad \text{and} \\ &\rho = \frac{\hat{\boldsymbol{x}}^{\mathrm{T}} \hat{\boldsymbol{y}}}{|\hat{\boldsymbol{x}}||\hat{\boldsymbol{y}}|} . \end{aligned} \tag{18}$$

In (18) we use s_j and $\hat{s}_j$ to denote the scalar components of the vectors $\boldsymbol{s}_i$ and $\hat{\boldsymbol{s}}_i$ respectively, and c_{jj} to denote the diagonal scalar components of the matrix $\boldsymbol{C}_i$. In the following we use the notation $[a_i,\ b_i] = \mathrm{DE}\left(\boldsymbol{s}_i, \hat{\boldsymbol{s}}_i, r_i, \boldsymbol{C}_i\right)$ to refer to the entire operation of the DE algorithm as described in (15), (16), (17), and (18). Since C_i must be normalized to satisfy (15), the DE algorithm can weight the data points within a data set, but it cannot weight the data points between data sets. Weighting between data sets is accomplished in a different step of the iterated nested least-squares algorithm described below.

2.4 Removing Excess Degrees of Freedom

As defined so far, our problem has two excess degrees of freedom (one scaling and one shifting) and this precludes a unique solution. This situation is easily remedied by constraining a_j=1, b_j=0 for some value of $1 \leq j \leq m$. This means that for the j^{th} data set, the first-order correction to the data is null. Thus the multiple data set fitting algorithm will transform all other data to the scale of the j^{th} data set. For this reason we refer to the j^{th} data set as the reference data set. For simplicity, and without loss of generality, we will assume j=1, so that the first data set is the reference data set. The constraints a_1=1, b_1=0 can be enforced by shifting and scaling $\boldsymbol{w}, \{a_i\}_{i=1}^m$, and $\{b_i\}_{i=1}^m$ at each iteration of the multiple data set fitting algorithm:

$$
\begin{aligned}
\tilde{w}_i &= w_i / a_1, & i &= 1 \text{ to } r,\\
\tilde{w}_{r+1} &= \left(w_{r+1} - b_1\right) / a_1, & &\\
\tilde{b}_i &= \left(b_i - b_1\right) / a_1, & i &= 1 \text{ to } m,\\
\tilde{a}_i &= a_i / a_1, & i &= 1 \text{ to } m.
\end{aligned}
\tag{19}
$$

2.5 The Iterated Nested Least-Squares (INLS) Algorithm

We are now equipped to return to the original problem of fitting parameters and scores. We select initial values of $\{a_i\}_{i=2}^{m}$ and $\{b_i\}_{i=2}^{m}$ using prior knowledge of the subjective testing scales used in the m data sets. One simple and intuitive rule for selecting these initial values of a_i and b_i is that the resulting first order correction should map the endpoints of the subjective test scale used in the i^{th} data set to the endpoints of the subjective test scale used in the reference data set. Once this initialization is completed, the INLS algorithm has three major steps per iteration. First it solves the least squares problem given in (8) yielding $\boldsymbol{w}$. Next it uses the DE algorithm to find $\{a_i\}_{i=1}^{m}$ and $\{b_i\}_{i=1}^{m}$ consistent with (13) and (14). Finally the normalization procedure given in (19) is applied. These three steps are repeated until convergence criteria on one or more of $\{a_i\}_{i=1}^{m}$, $\{b_i\}_{i=1}^{m}$ or $\tilde{\boldsymbol{w}}$ are satisfied. We offer no mathematical proof that this iterative approach will always converge, but we do note that in each of our actual applications it has always converged.

The INLS algorithm for fitting multiple data sets is summarized below. When necessary, the notation developed so far is augmented with an iteration number. For example, $\tilde{\boldsymbol{s}}_i(j)$ is the value of $\tilde{\boldsymbol{s}}_i$ in the j^{th} iteration of the algorithm.

Inputs:

$\{\boldsymbol{s}_i\}_1^m$, score vectors

$\boldsymbol{P}$, parameter matrix

$\boldsymbol{C}$, cost matrix

$\{r_i\}_1^m$, cost-weighted error power ratios

$\{\tilde{a}_i(0)\}_{i=1}^{m}$, $\{\tilde{b}_i(0)\}_{i=1}^{m}$, initial data set correction factors

Local Variables:

i, data set number

j, iteration number

$\{\tilde{a}_i(j)\}_{i=1}^{m}$, $\{\tilde{b}_i(j)\}_{i=1}^{m}$, current data set correction factors

$\boldsymbol{w}(j)$, current parameter weights before normalization step

$\tilde{\boldsymbol{w}}(j)$, current parameter weights after normalization step

$\tilde{\boldsymbol{s}}_i(j)$, current corrected score vectors, per data set

$\tilde{\boldsymbol{s}}(j)$, current corrected score vectors, all data sets

$\hat{\boldsymbol{s}}_i(j)$, current parameter-based approximations to corrected score vectors, per data set

$\hat{\boldsymbol{s}}(j)$, current parameter-based approximations to corrected score vectors, all data sets

Extract the *m* per data-set cost matrices $\{\boldsymbol{C}_i\}_{i=1}^{m}$ from $\boldsymbol{C}$ and normalize each per data-set cost matrix so that the squared diagonal entries sum to one: $\sum_{j=1}^{n_i} c_{jj}^2 = 1$

Algorithm:

```
$j$=0
While convergence criteria on $\tilde{\boldsymbol{w}}$ , and/or $\{\tilde{a}_i\}_{i=2}^{m}$ , and/or $\{\tilde{b}_i\}_{i=2}^{m}$ are not satisfied
    $j$=$j$+1
    $\tilde{\mathbf{s}}_i(j)=\tilde{a}_i(j-1)\mathbf{s}_i+\tilde{b}_i(j-1)\mathbf{1}$ , $i=1$ to $m$
    $\tilde{\boldsymbol{s}}(j)^{\mathrm{T}} = \left[\tilde{\boldsymbol{s}}_1(j)^{\mathrm{T}} \ ,\tilde{\boldsymbol{s}}_2(j)^{\mathrm{T}},\tilde{\boldsymbol{s}}_3(j)^{\mathrm{T}}, \ \ldots \ , \ \tilde{\boldsymbol{s}}_m(j)^{\mathrm{T}}\right]$
    $\boldsymbol{w}(j) = \left(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{C}^{\boldsymbol{2}}\boldsymbol{P}\right)^{-1}\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{C}^{\boldsymbol{2}}\tilde{\boldsymbol{s}}(j)$
    $\hat{\boldsymbol{s}}(j) = \boldsymbol{Pw}(j)$
    Extract $\{\hat{\boldsymbol{s}}_i(j)\}_{i=1}^{m}$ from $\hat{\boldsymbol{s}}(j)$ , consistent with
        $\hat{\boldsymbol{s}}(j)^{\mathrm{T}} = \left[\hat{\boldsymbol{s}}_1(j)^{\mathrm{T}} \ ,\hat{\boldsymbol{s}}_2(j)^{\mathrm{T}},\hat{\boldsymbol{s}}_3(j)^{\mathrm{T}}, \ \ldots \ , \ \hat{\boldsymbol{s}}_m(j)^{\mathrm{T}}\right]$
    $[a_i(j), \ b_i(j)] = DE\left(\boldsymbol{s}_i,\hat{\boldsymbol{s}}_i(j),r_{i,}\boldsymbol{C}_i\right)$ ,        $i = 1$ to $m$
    $\tilde{b}_i(j) = \left(b_i(j)-b_1(j)\right)/a_1(j)$ ,        $i = 1$ to $m$
    $\tilde{a}_i(j) = a_i(j)/a_1(j)$ ,        $i = 1$ to $m$
    $\tilde{w}_i(j) = w_i(j)/a_1(j)$ ,        $i = 1$ to $r$
    $\tilde{w}_{r+1}(j) = \left(w_{r+1}(j)-b_1(j)\right)/a_1(j)$
```

```
End
$\tilde{\boldsymbol{s}}_i(j) = \tilde{a}_i(j)\ \boldsymbol{s}_i + \tilde{b}_i(j)\boldsymbol{1}$ ,        $i = 1$ to $m$
$\hat{\boldsymbol{s}}(j) = \boldsymbol{P}\tilde{\boldsymbol{w}}(j)$

Outputs:
    $\{\tilde{a}_i(j)\}_{i=1}^{m}$, $\{\tilde{b}_i(j)\}_{i=1}^{m}$, final data set correction factors
    $\tilde{\boldsymbol{s}}(j)$, corrected score vector
    $\tilde{\boldsymbol{w}}(j)$, parameter weights
    $\hat{\boldsymbol{s}}(j)$, parameter-based approximation to corrected score vector
```

In our applications, the INLS algorithm has always converged in fewer than twenty iterations. A typical convergence criterion would be that between two iterations, the largest relative magnitude change in any of $\{\tilde{a}_i\}_{i=2}^{m}$, $\{\tilde{b}_i\}_{i=2}^{m}$, or any component of $\tilde{\boldsymbol{w}}$ must be less than some threshold:

$$\max\left(\max_i\left(\frac{\left|\tilde{a}_i(j)-\tilde{a}_i(j-1)\right|}{\left|\tilde{a}_i(j-1)\right|}\right),\ \max_i\left(\frac{\left|\tilde{b}_i(j)-\tilde{b}_i(j-1)\right|}{\left|\tilde{b}_i(j-1)\right|}\right),\ \max_i\left(\frac{\left|\tilde{w}_i(j)-\tilde{w}_i(j-1)\right|}{\left|\tilde{w}_i(j-1)\right|}\right)\right) < \Delta\,. \tag{20}$$

The INLS algorithm may seem somewhat indirect when it comes to enforcing the constraints a_1=1, b_1=0. A seemingly more direct approach would be to select a_1=1, b_1=0 for the initial values, and never apply the DE algorithm to the scores from the reference data set. This variation of the INLS algorithm eliminates the need for the normalization steps described in (19). This variation leaves the scores from the reference data set anchored at their original location, so we call this the anchored iterated nested least-squares (AINLS) algorithm. This algorithm converges much more slowly than the INLS algorithm. When skipping the DE step on the scores from the reference data set, direct information about the relationship between the reference data set scores and the parameters is ignored. This information is apparently extracted indirectly and less efficiently through the scores of the remaining m-1 data sets, leading to slower convergence. Further, in data sets using simulated scores and parameters, we have found the INLS algorithm is much more likely to converge to results closer to the known correct values of $\{\tilde{a}_i\}_{i=2}^{m}$, $\{\tilde{b}_i\}_{i=2}^{m}$, and $\tilde{\boldsymbol{w}}$ than the AINLS algorithm. Thus we will not discuss the AINLS algorithm any further in this memorandum.

3. EXAMPLE ALGORITHM RESULTS

3.1 Simulated Data Results

In Figures 2-6 we show the operation of the INLS algorithm on simulated scores and parameters. This simulation includes r=3 parameters and m=3 data sets. Each data set contains n_i=40 data points for a total of n=120 data points. We simulated parameters by limiting uniformly distributed random variables to appropriate ranges and simulated the corresponding scores by linearly combining parameters and adding Gaussian noise. Thus we know that the "right answers" for this simulation are $\tilde{a}_2 = 2.0$, $\tilde{a}_3 = 0.9$, $\tilde{b}_2 = -1.0$, $\tilde{b}_3 = -0.7$, and $\tilde{\boldsymbol{w}} = \begin{bmatrix}0.2 & 0.4 & 0.8 & 0.4\end{bmatrix}^T$. Since the noise was added only to the scores, we set $r_i^2 = 10$ for all three data sets to reflect this limiting case. We use uniform cost-weighting on all data sets:

$$\boldsymbol{C}_i = \frac{1}{\sqrt{40}} \boldsymbol{I}_{40\times 40}\,. \tag{21}$$

Figures 2-4 are scatter plots of the corrected scores $\tilde{\boldsymbol{s}}$ vs. the parameter-based approximations to corrected scores $\hat{\boldsymbol{s}}$. In these figures, the data points from data set 1 (the reference data set) are in blue, those from data set 2 are green, and those from data set 3 are in red. Figure 2 shows the relationship between the scores and $\hat{s}$ before the INLS algorithm has been started. We calculate this initial value of $\hat{\boldsymbol{s}}$ using (2). Note that each set of scores shows a correlation to $\hat{\boldsymbol{s}}$, yet the data points of data sets 2 and 3 do not align with the data points of data set 1. Figure 3 shows these same relationships after one iteration of the INLS algorithm, and Figure 4 shows these relationships after two iterations. Note that the three groups of data points now align well. The relationship shown in Figure 4 does not change significantly with further iterations of the INLS algorithm. The evolutions of $\tilde{a}_2, \tilde{a}_3, \tilde{b}_2, \tilde{b}_3,$ and $\tilde{\boldsymbol{w}}$ are shown in Figure 5. Each of these converges to a value at or near the known "right answers." Finally, Figure 6 shows how the root mean-squared error (*RMSE*) between $\tilde{\boldsymbol{s}}$ and $\hat{\boldsymbol{s}}$ decreases as the INLS algorithm iterates. We define *RMSE* as

$$RMSE = \sqrt{\frac{1}{n}\left|\tilde{\boldsymbol{s}} - \hat{\boldsymbol{s}}\right|^2}\,. \tag{22}$$

From these figures it is clear that in this example, the INLS algorithm does virtually all of the fitting that it will ever do in the first two algorithm iterations.

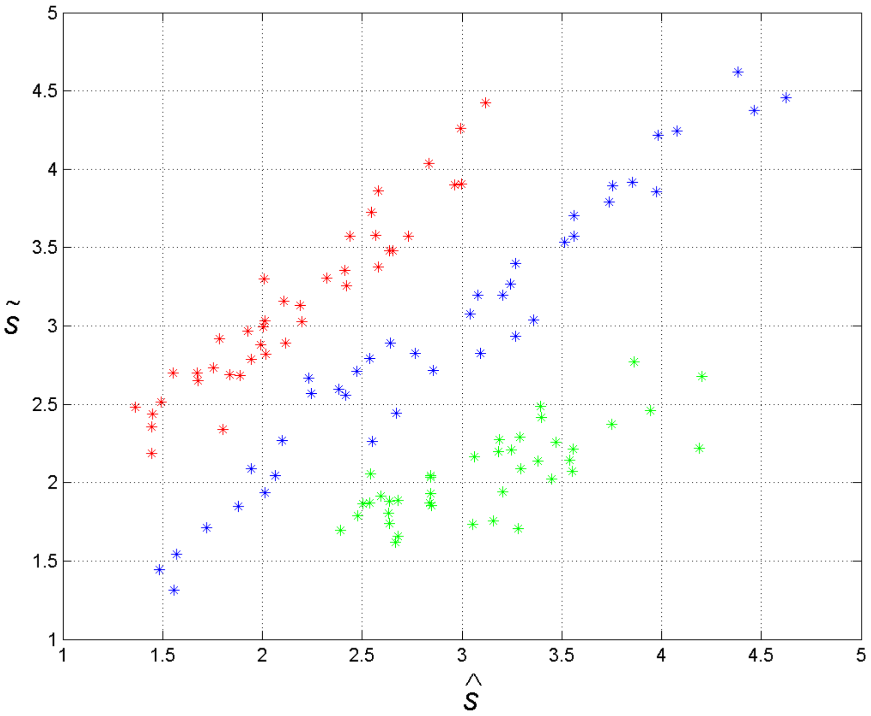


Figure 2. Simulated data input to INLS algorithm; each color represents one data set.

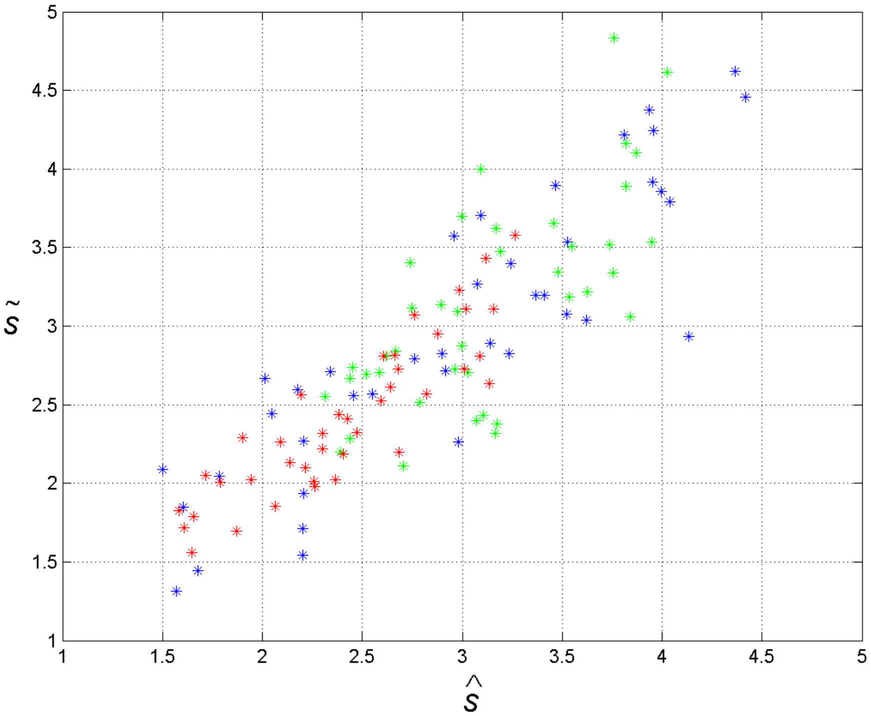


Figure 3. Data of Figure 2 after one iteration of INLS algorithm.

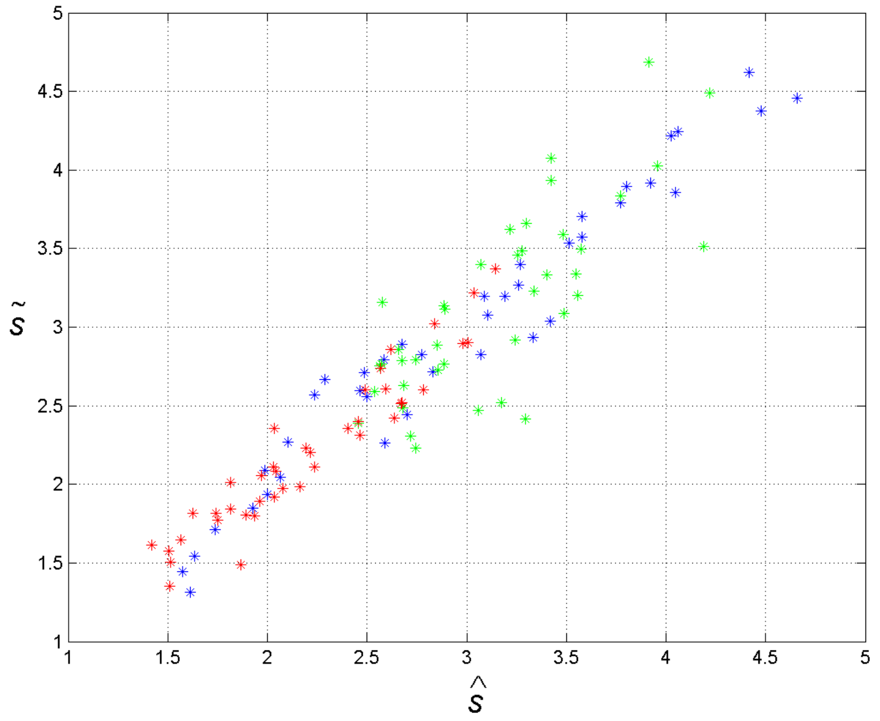


Figure 4. Data of Figure 2 after two iterations of INLS algorithm.

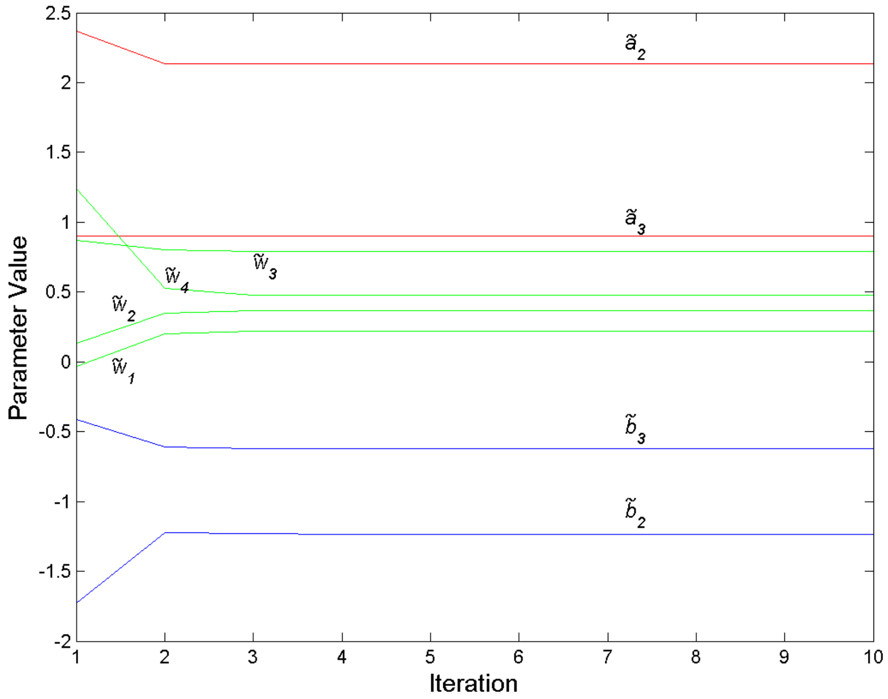


Figure 5. Evolution of values as INLS algorithm iterates on data of Figure 2.

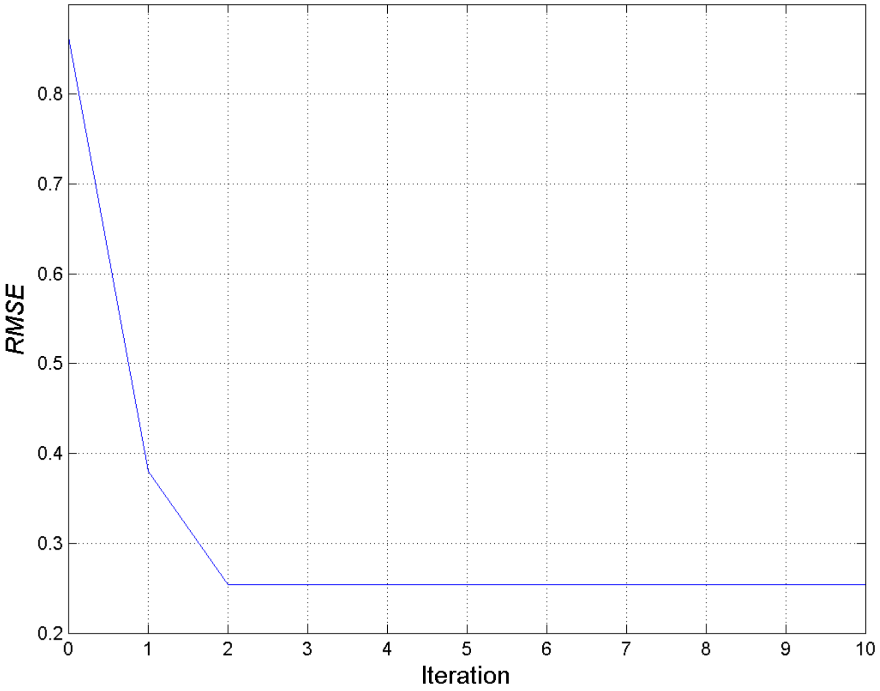


Figure 6. Evolution of *RMSE* as INLS algorithm iterates on data of Figure 2.

3.2 Actual Data Results

In Figures 7-11 we show the operation of the INLS algorithm on actual scores and parameters. We use r=13 parameters taken from a measuring normalizing block algorithm for estimating telephone-bandwidth speech quality [3]. We have included m=3 data sets. In each data set, the subjective test was an absolute category rating test that used the five-point mean opinion score scale [11]. Together these data sets contain over 3000 data points and the resulting scatter plots are not readable. For demonstration purposes here we have randomly selected n_i=40 data points from each data set for a total of n=120 data points. Since we have no prior knowledge of relative error sources in this data, we choose to assume that the parameter error and the score error are equal and thus we set r^2=1 for all three data sets. We use uniform cost-weighting on all data sets as given in (21).

Figures 7-9 are scatter plots of the corrected scores $\tilde{s}$ vs. the parameter-based approximations to the corrected scores $\hat{s}$. These figures use the same color coding as the previous set of figures. Figure 7 shows the relationship between the scores and $\hat{s}$ before the INLS algorithm has been started. We calculate this initial value of $\hat{s}$ using (2). Each set of scores shows a correlation to $\hat{s}$, and any further relationships are much more subtle than in the simulated data. Careful visual inspection does reveal that both the green and red data points should be scaled by a scale factor that is less than one. That is, the slopes of the green and red data clouds should be reduced in order to best line up with the slope of the blue data cloud. In other words, we expect $0 < \tilde{a}_2 < 1$ and $0 < \tilde{a}_3 < 1$. Further, it appears that after such scaling, the green data points should be shifted up by a small amount to best agree with the blue data points, and the red data points may require a smaller upward shift or perhaps no upward shift (i.e., $0 < \tilde{b}_3 < \tilde{b}_2 < 1$).

Figure 8 shows the $\tilde{s}$ vs. $\hat{s}$ scatter plot after one iteration of the INLS algorithm, and Figure 9 shows the same after five iterations. Note that the three groups of data points now align better than before. The relationship shown in Figure 9 does not change significantly with further iterations of the INLS algorithm. The evolutions of $\tilde{a}_2$, $\tilde{a}_3$, $\tilde{b}_2$, and $\tilde{b}_3$ are shown in Figure 10 and each converges to a value consistent with our initial visual assessment. (Since $\tilde{\boldsymbol{w}}$ contains 14 elements, plots of that vector are not easily readable and we have elected not to plot it.) Finally, Figure 11 shows how the root mean-squared error (*RMSE*) between $\tilde{s}$ and $\hat{s}$ evolves as the INLS algorithm iterates. Note that the RMSE is minimized after five iterations and remains nearly unchanged beyond that point. On the other hand, the values of $\tilde{a}_2$, $\tilde{a}_3$, $\tilde{b}_2$, $\tilde{b}_3$, and $\tilde{w}$ continue to evolve until about 15 iterations have passed. This means that the INLS algorithm has found a family of about ten similar solutions that are almost equally desirable in terms of *RMSE*. We have run 100 iterations on this data and have verified that the solution remains stable and that there is no further increase in *RMSE*.

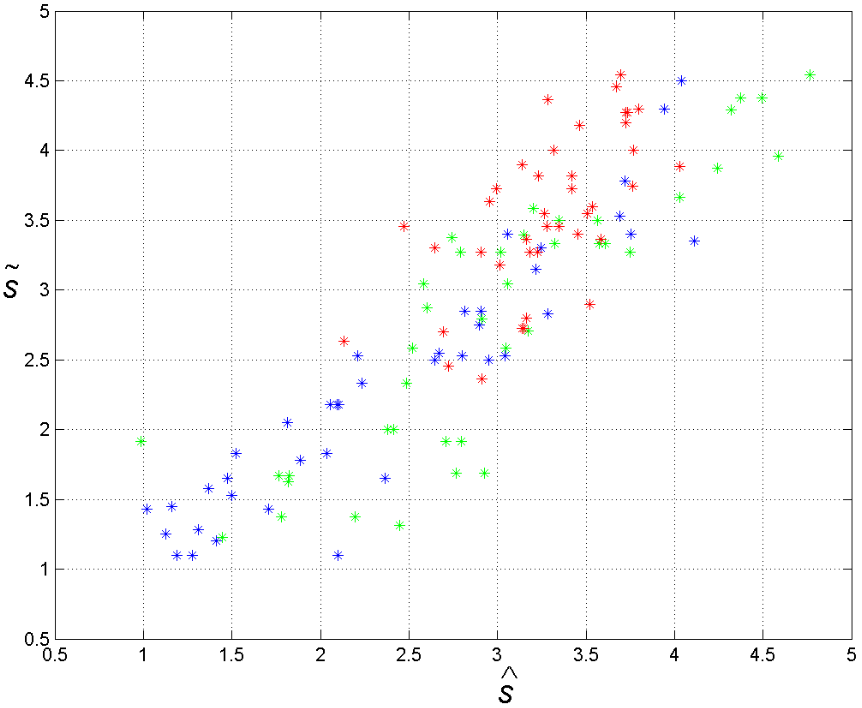


Figure 7. Speech quality data input to INLS algorithm; each color represents one data set.

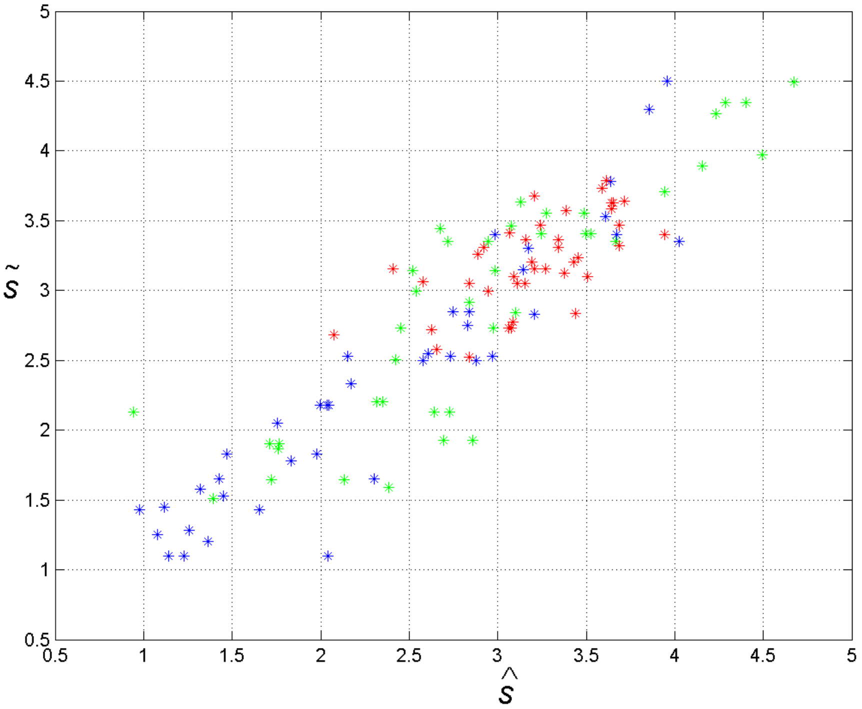


Figure 8. Data of Figure 7 after one iteration of INLS algorithm.

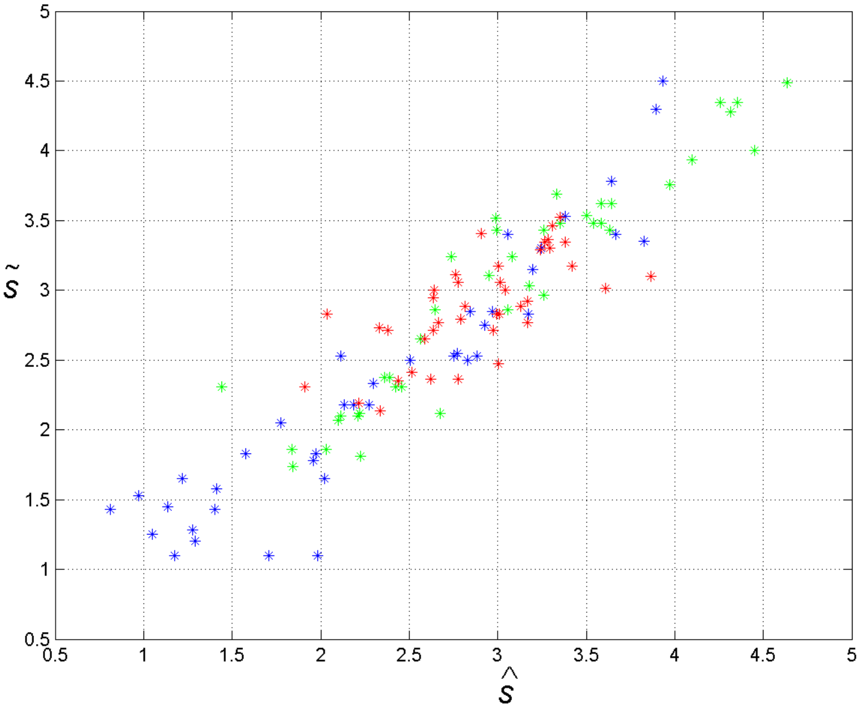


Figure 9. Data of Figure 7 after five iterations of INLS algorithm.

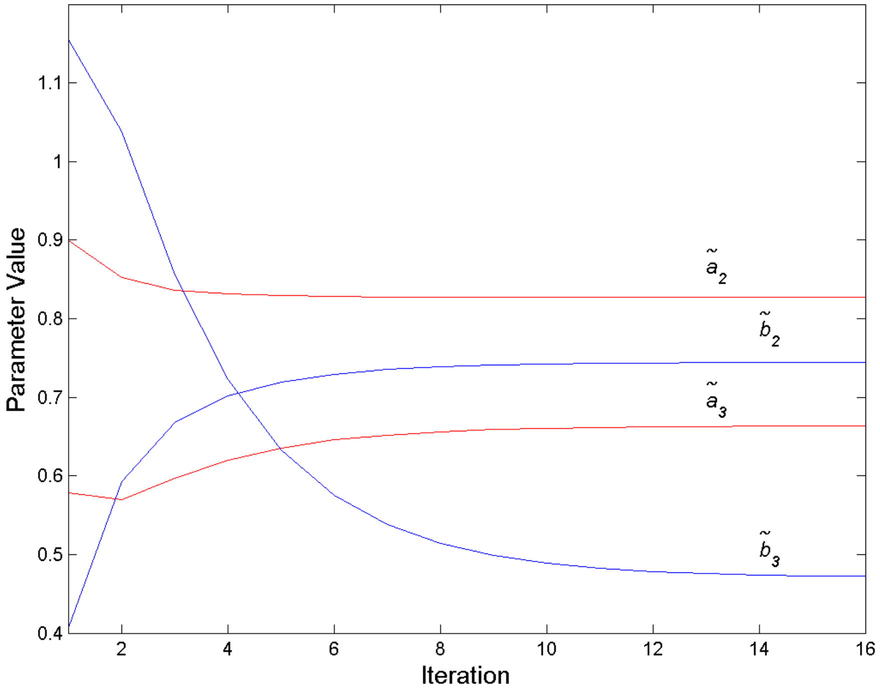


Figure 10. Evolution of values as INLS algorithm iterates on data of Figure 7.

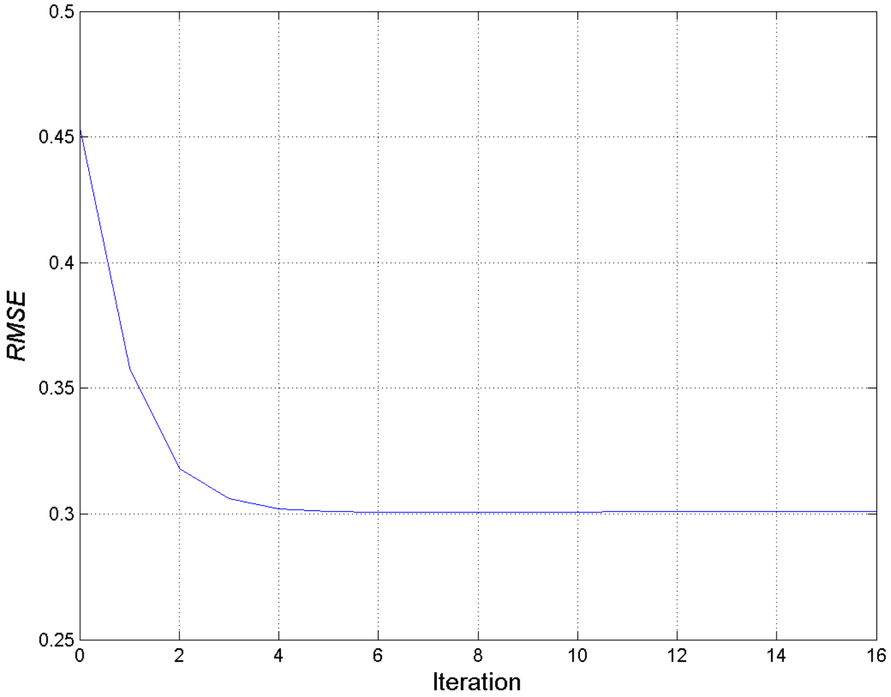


Figure 11. Evolution of *RMSE* as INLS algorithm iterates on data of Figure 7.

4. SUMMARY

We have provided motivation for the multiple data set fitting scenario described in Figure 1. While our motivation comes from the development of objective estimators of perceived audio and video quality, our solutions are general and will apply whenever the diagram in Figure 1 applies. Our solutions are the INLS and AINLS algorithms, and the INLS algorithm is clearly the preferred algorithm. We have presented example results for the INLS algorithm operating on both simulated and real data. The INLS algorithm finds individual first-order corrections for the score vectors in each data set. It also finds a single weight vector that forms a linear combination of the parameters across all data sets. The INLS reduces the *RMSE* calculated across all data sets as it aligns disparate data sets with each other to form a single cluster of data.

5. REFERENCES

[1] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra, "Perceptual evaluation of speech quality (PESQ) – A new method for speech quality assessment of telephone networks and codecs," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing 2001*, Salt Lake City, May 2001, vol. 2, pp. 749-752.

[2] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," Geneva, 2001.

[3] S.D. Voran, "Objective estimation of perceived speech quality, part I: Development of the measuring normalizing block technique," *IEEE Transactions on Speech and Audio Processing,* vol. 7, no. 4, pp. 371-382, Jul. 1999.

[4] S.D. Voran, "Objective estimation of perceived speech quality, part II: Evaluation of the measuring normalizing block technique," *IEEE Transactions on Speech and Audio Processing,* vol. 7, no. 4, pp. 383-390, Jul. 1999.

[5] ITU-T Recommendation P.861, "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs," Geneva, 1996.

[6] T. Thiede, W.C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J.G. Beerends, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, and B. Feiten, "PEAQ-The ITU standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, vol. 48, no. 1/2, Jan./Feb. 2000.

[7] ITU-R Recommendation BS.1387, "Method for objective measurements of perceived audio quality," Geneva, 2001.

[8] S. Wolf, "Measuring the end-to-end performance of digital video systems," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 320-328, Sep. 1997.

[9] S. Wolf and M. Pinson, "Video quality measurement techniques," NTIA Report 02-392, Jun. 2002 (available at http://www.its.bldrdoc.gov/n3/video/documents.htm).

[10] C.J. van den Branden Lambrecht, D.M. Costantini, G.L. Sicuranza, and M. Kunt, "Quality assessment of motion rendition in video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 766-782, Aug. 1999.

[11] ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality," Geneva, 1996.

[12] L.V. Hedges and I. Olkin, *Statistical Methods for Meta-Analysis*, Orlando, FL: Academic Press, 1985.

[13] S.D. Voran, “Estimation of system gain and bias using noisy observations with known noise power ratio,” NTIA Report 02-395, Sep. 2002.