Songklanakarin J. Sci. Technol.
36 (2), 241-248, Mar. - Apr. 2014

SONGKLANAKARIN JOURNAL OF SCIENCE AND TECHNOLOGY
http://www.sjst.psu.ac.th

Original Article

A practical pedestrian approach to parsimonious regression with inaccurate inputs

Seppo Karrila*

Faculty of Science and Technology,
Prince of Songkla University, Pattani Campus, Mueang, Pattani, 94000 Thailand.

Received 27 December 2011; Accepted 4 December 2013

Abstract

A measurement result often dictates an interval containing the correct value. Interval data is also created by round-off, truncation, and binning. We focus on such common interval uncertainty in data. Inaccuracy in model inputs is typically ignored on model fitting. We provide a practical approach for regression with inaccurate data: the mathematics is easy, and the linear programming formulations simple to use even in a spreadsheet. This self-contained elementary presentation introduces interval linear systems and requires only basic knowledge of algebra. Feature selection is automatic; but can be controlled to find only a few most relevant inputs; and joint feature selection is enabled for multiple modeled outputs. With more features than cases, a novel connection to compressed sensing emerges: robustness against interval errors-in-variables implies model parsimony, and the input inaccuracies determine the regularization term. A small numerical example highlights counterintuitive results and a dramatic difference to total least squares.

Keywords: compressed sensing, joint sparsity, matrix uncertainty, interval linear systems, robust regression

1.Introduction

It is common to use measured and therefore inaccurate predictive features while identifying (fitting the parameters of) a regression model and it is also common to ignore their inaccuracy. Ordinary regression analysis assumes accurate inputs (the independent variables) and assigns all inaccuracy to the output (the dependent variable). For noisy inputs, there is total least squares (TLS), but recent significant developments by Wiesel et al. (2008) demonstrate that the statistical tools and techniques for errors-in-variables are not yet perfected.

One inherent flaw of TLS is particularly easy to see. A perfect fit is possible with a full rank square system, or with an underdetermined system having a lot of candidate predictors, and this fit is optimal for the criterion minimized. This means that the inaccuracies of predictors are completely ignored, however large they might be. But a human modeler will avoid giving much weight to a comparatively inaccurate predictive feature, because in production use of the model that inaccuracy will be passed on to the predicted output. We desire and construct an alternative method that, in this respect, agrees with the modeler.

Also, the TLS approach assumes the inaccuracies are Gaussian normal, so that maximum likelihood is achieved by minimizing the sum of squared errors in both inputs and output, scaled to equal variances.

However, in technical applications this Gaussian assumption will often be violated: the measurement errors will never be very large, not even with a small but non–zero probability. For example, a weighing scale might be accurate within ±5% (proportional inaccuracy) or within 0.1 kg (uniform inaccuracy). A realistic more complicated scenario is that the accuracy will be 0.1 kg for weights up to 50 kg, and 0.2 kg for higher weights up to the maximum of range.

* Corresponding author.
Email address: seppo.karrila@gmail.com,
seppo-k@bunga.pn.psu.ac.th

While such characterizations are not a prominent part of traditional statistics, they are quite common in measurement technology. Engineering design benefits from performance guarantees, and pursuing them is reasonable in deterministic macroscopic systems: measurement devices come with guarantees enforced in quality control, and their usage protocols include regular calibration.

The above deterministic measurements give finite intervals that contain the correct value, and we have no probabilistic assumptions about the distribution of errors. It is perfectly natural for a measurement to have a bias, so that a 5% accurate scale may consistently show 3% too high values. In the case of round-off to nearest integer, the error can be anything from -0.5 to +0.5, and other operations that generate interval inaccuracy include truncation, binning, and quantization of any sort. In practice these are sometimes combined, so that the reading from a 5% accurate scale is further rounded, for example, to two digits on recording it. In all these cases, an interval containing the true value is still easily computable from the recorded value.

Since interval inaccuracy is obviously very common, we choose to pursue it, and only this one in this paper. We will provide a simple deterministic approach in using inaccurate predictive features, such that guarantees robustness against interval uncertainty of the data; the model performance suffers from the uncertainty as little as possible. The above types of measurement inaccuracy will actually be easily dealt with.

Various number counts can be fully accurate integer values, and are simply represented by intervals of zero width: the lower and upper bounds coincide. Therefore accurate values need no separate treatment in the context of interval inaccuracy.

The predictors constructed are linear models, so we perform linear regression, although not by least squares. Since we do not know where each correct value is in its interval, we require that for ANY value in each interval the predictor gives closely similar output to the targeted values. This means minimizing the worst case output error, for all values allowed by the interval uncertainties.

In the language of robust optimization, the intervals for model inputs define an uncertainty set, specifically of box type in the terminology used by Ben-Tal *et al.* (2009) in their comprehensive book. Robust optimization could be used to solve this problem, and the book referred to has in its exercises some results on box-type (i.e., interval) uncertainty. However, these results are not accessible to many practical modelers, who are not familiar with conic duality and convex optimization, and it is delightful that a much less demanding pedestrian approach works very well and is transparent.

The work of El Ghaoui and Lebret (1997) on robust modeling requires even more advanced background in mathematics, on linear fractional matrix transforms, and is only accessible to specialists. The tools needed for second order cone programming or semi-definite programming are not widely accessible either, but the theory leads to these optimization problems.

The book by Tarantola (2005) appears to not give ready-to-use recipes, but deals in a general setting with topics related to also our work. Its flavor is more philosophical and conceptual than for a practitioner who needs a simple solution to an apparently simple problem.

The impression from prior literature then is that while robust modeling that acknowledges data inaccuracy has been practiced, it has been loaded with heavy theory that keeps it inaccessible to all but select specialists, who must invest heavily in studying the relevant topics. Our contention is that simplified approaches and techniques are needed for the non-specialists who have to fit models to inaccurate data, so that the most frequent input inaccuracy types become manageable to a typical student of science or engineering. This is exactly what the current work provides.

We will emphasize sparsity, meaning that only a few of the available candidate features are used in the eventual model. In essence, we perform “feature selection” to achieve “model parsimony”. Our results on sparsity are not implied by the general theory of robust optimization, but are an original observation that links this work to the general field of compressed sensing.

Prior published work such as Xu *et al.* (2009) or Rosenbaum and Tsybakov (2008), that relate to both sparsity and inaccurate inputs, have restrictive assumptions made about the inaccuracies in the predictive features, or constrain the parameters solved for. In contrast, the current work allows completely arbitrary finite intervals for each and every measured value, including also accurate values with the intervals shrunk to points, and the model parameters are not constrained. However, linear inequality constraints can be added at will, for example if a model parameter is known to be positive.

In multiple regression statisticians have practiced feature selection, to include in the final model only those predictive input features that are useful (statistically significant) and to exclude such candidate features that appear irrelevant to the prediction task at hand. Forward selection adds in sequence the most useful new feature to the model, while backward elimination starts by including all of them and shaves off the least relevant ones step by step. More complicated dances that take steps both forward and backward can be found in the literature; the reason has been the desire to improve stepwise selection methods that are known to work less than perfectly. However, all these schemes face an inherent problem and will remain imperfect. They are greedy approaches trying to deal with a combinatorial problem: selecting the best predictive subset of features. None can guarantee that the best set is found. Furthermore, even with a small number of available features, the problem of how many predictive features to keep is not simple when based on assumptions commonly made in statistics, as illustrated by Akaike or Bayes information criteria, usually referred to simply as AIC and BIC. These criteria help compare alternative models identified from the same data, but the scores computed for a single individual model are neither informative nor interpretable. The interested reader can pursue these topics in great detail with the book by Hastie *et al.* (2009), where the approach assumes Gaussian errors.

In contrast, our approach selects the most robust feature set, and provides a concrete diagnostic of the model fit: the tolerance parameter. Too large tolerance compared to the targeted output values indicates failure of the modeling with current data: there is no such robust model that would, within the allowed input uncertainty, always give an acceptably good fit (i.e., not too large tolerance) with the targeted output values. Having such a clear diagnostic, not requiring artistic interpretation based on experience, is of high value to practical modelers. There will always be cases worth a try despite only a small amount of inaccurate data, and success and failure must be distinguished in such cases.

Also in contrast to statistical feature selection, the formulated linear programs are not solved by a greedy approximation, but by a reliable technology that finds the best solution; and this solution encompasses feature selection. The solution is best in the sense of minimizing worst case output error, within the uncertainty intervals of given data. We note though that examples with multiple (i.e., non-unique) optimal solutions can be constructed, though this case occurs rarely in linear programming (LP). A duplicate column in the design matrix A is an example that can easily be created accidentally, but this does not prevent LP from finding one of the solutions. However, it will make the typical formula for least squares unusable since A^TA is not invertible. As for that "reliable technology", we will simply assume the use of an "LP solver", and that it provides an optimal solution that could be found with the Simplex method, even if actually an interior point method may be used by a modern numerical solver. This LP approach deals handily with an excess of available features, in other words an underdetermined problem. We will show that the sparsity of solutions can be controlled with a simple trick, and now we are in the domain of compressed sensing, finding sparse solutions to underdetermined linear problems.

We will address the challenging problem of joint feature selection, also known as the MMV problem (multiple measurement variables). In this problem, we select one single set of features that enables predicting several outputs: doing feature selection separately for each targeted output is not good enough. This is in the general domain of multi-task learning where benefits are expected if the outputs are related, so that the simultaneous learning/modeling tasks support each other. The approach given is original and quite different from prior approaches in the literature, to which some key references are given in the Section Discussion.

Each of the problem domains above, namely inaccuracy of predictive features, feature selection, multi-task learning, and compressed sensing, has its own body of literature that we will not try to review. However, the references given cover more than the intersection of these domains. We will simply pursue a novel and practically doable algorithmic approach that, within its limitations, does handle all of these problems in any of their combinations. The limitations will be elaborated in the Section Discussion, where our theoretical results are also briefly compared to recent closely related publications.

2.Theory

Consider a linear model $B = AX$ where each of the n rows in the given B and A represents a single case. The p columns of A each represent a measured (candidate) predictive feature, while the p rows of unknown X provide weights (model parameters) that will be identified by fitting the model, so it approximates the targeted output values in B. Ordinarily B and X would be column vectors in multiple linear regression, but in our case there are m columns in each, making this an MMV problem. Without requiring joint feature selection, those m columns could each be treated separately. In statistics A is called the design matrix as it reveals the experimental design of controlled inputs, while in compressed sensing it is called the dictionary and its columns the atoms.

With an accurate numeric design matrix A this would be the common MMV problem, but we allow inaccuracy in A as follows. Each element in the interval matrix $[A]$ is a finite interval, and the lower bounds are collected in matrix A_{min} while the upper bounds define A_{max}. The inequality $A_{min} \le \tilde{A} \le A_{max}$ holds elementwise for any realization $\tilde{A} \in \left[A\right] = [A_{min}, A_{max}]$ Think of the realization as a possible combination of true values based on inaccurate measurements.

So brackets around a single symbol can be read as "interval", and brackets around two comma-separated symbols mean "interval from … to …". Our finite intervals are always closed, and the corresponding inequalities are not strict but allow also equality to the bound.

We define the center $\breve{A} = (A_{min} + A_{max}) / 2$ and the elementwise nonnegative radius $rad(A) = (A_{max} - A_{min}) / 2$, and use these to denote an interval matrix also by $[A]$ = $\left[\breve{A} \pm rad\left(A\right)\right]$ to explicitly show the center and radius. The symbol "±" within the brackets identifies this case of notation.

Similar notations are used also with vectors, as they are just skinny matrices: an interval vector can also be specified by its bounds, or by its center and radius.

So the interval matrix $[A]$ is actually a set of matrices, whose elements independently can take values in their respective allowed ranges. Fixing these values gives a single representative or realization, an ordinary matrix $\tilde{A} \in \left[A\right]$.

As an example, the numbers in matrix $\breve{A} = \begin{pmatrix} 0.7 & 12 \\ 5 & 0.3 \end{pmatrix}$ are rounded off, so that the inaccuracy is given by the radius

$rad(A) = \begin{pmatrix} 0.05 & 0.5 \\ 0.5 & 0.05 \end{pmatrix}$. The corresponding interval matrix is $[A] = [\breve{A} \pm rad(A)]$, and the 1,1-element of a representative can be anything from 0.65 to 0.75.

Arithmetic operations with [*A*] are immediately defined by recognizing it as a set, and the outcome of an operation is the set collecting outcomes for all realizations $\tilde{A} \in [A]$. However, in practice the effect of an interval matrix on multiplying an accurate numeric vector x is best expressed in terms of the center and radius of the outcome: $[A]x =$ $[\breve{A}x \pm rad(A)|x|]$. The corresponding upper and lower bounds can be expressed by using the positive and negative parts of vector *x*, and are shown in the linear program formulations later. Note that the absolute value of *x* is needed in the resulting radius. This causes the inaccuracies to add up for the worst case, instead of canceling each other with different signs.

The conversion to the form used in the LP formulations is basic algebra, using the facts that $x = x_+ - x_-$ and $|x| = x_+ + x_-$, where the positive part x_+ has the negative components of *x* replaced by zeroes, and the negative part x_- is the positive part of $-x$.

It is uncommon to calculate with intervals, as inequalities are more difficult to handle than ordinary equalities/equations, and an interval is just a combination of two inequalities. The algebra of computing with intervals is known as interval analysis, and the book by Moore (1966) provides an introduction to it. However, we do not need specific results from it.

While a linear system of equations $Ax = b$ of reasonable size is routinely solved, it is not obvious what is meant by a solution to the interval equations "$[A]x = b$", where *b* is the targeted output of our linear model and [*A*] holds our inaccurate feature values. The interval vector on left cannot equal the accurate numeric vector on right unless *x* and *b* are both zeroes. We have to expand also *b* to an interval vector.

We choose to require that $[A]x \subseteq [b]$, and note that for scalar intervals $[\alpha] \subseteq [\beta]$ if, and only if $|\breve{\alpha} - \breve{\beta}| \le rad(\beta) - rad(\alpha)$. This is easily seen by making a sketch on the real line, or considering when a smaller circle in plane fits within a larger one. Our condition for a solution becomes this component-wise inequality between two vectors: $|\breve{A}x - \breve{b}| + rad(A)|x| \le rad(b)$

Once *x* satisfies this condition, any realization $\tilde{A} \in [A]$ gives $\tilde{A}x \in [b]$. In other words, the identified model parameters *x* will map inaccurately replicated data $\tilde{A}$ also to $\tilde{b}$, not accurately but within tolerance $rad(b)$. This is exactly the kind of robustness we want of an identified linear model; if repeating the experiment would give drastically altered predictions for each repeated case, the model would be too sensitive to measurement noise in the predictive features. When the tolerance $rad(b)$ is given, the solution set for *x* is known as the tolerance solutions in the theory of linear interval systems (Fiedler *et al.*, 2006).

Clearly any *x* satisfies the condition above if we just make $rad(b)$ big enough, but for a useful model it should be small compared with *b* itself. So we will pursue the smallest possible $rad(b)$ and select the optimal *x* accordingly.

For convenience, we measure the size of $rad(b)$ by the largest absolute value of its elements, which is known as the sup-norm $\|rad(b)\|_\infty$. A linear program (LP) that minimizes this sup-norm (scalar denoted by *s*) is :

Minimize *s*

$$x_+ \ge 0, x_- \ge 0, s \ge 0$$

$$A_{max}x_+ - A_{min}x_- \le b + se$$

$$A_{min}x_+ - A_{max}x_- \ge b - se$$

where $e = (1, \ldots, 1)^T$ has all elements equal to one, and our optimal solution is $x^* = x_+ - x_-$.

This program solves for the parameters *x* that are most robust against the uncertainty within interval matrix [*A*], in the sense that *b* is reproduced as closely as possible for any matrix realization. The closeness is explicitly given by the numerical solution as the tolerance parameter *s* that limits deviation from targeted output for each and every component (i.e., for each case). Obviously [*A*] can include accurate elements, columns that have a uniform radius (same for each element), those that have proportional radii, and columns with no simple rule for the radii: mixtures of common types of measurement inaccuracy are covered by this approach.

LP problems have been numerically solved for over half a century, and their numerical treatment has developed into a reliable technology. For example, all common spreadsheet programs include LP solvers by default, capable of treating at least up to 200 unknowns. In the program above, the number of unknowns is 2*dim(*x*)+1, so at least tens of parameters can be identified in these basic and popular, commonly accessible software tools.

Having multiple vectors *b* with the same design or measurement matrix [*A*], we could apply the above LP separately to each. In case there are many more cases than parameters to identify, meaning that the matrix is tall and skinny (*n* > *p*), this is a quite reasonable approach.

Assume now that we have more candidate features than cases (*p* >> *n*), the matrix is fat with more columns than rows. Such situation is common for example in genomics, where tens of thousands of features are measured from each sample. The LP will provide a solution *x** with no more than *n* non–zero components, where *n* is the number of cases. However, unlike linear equation systems the LP is quite happy with many more unknowns than equations. We are able to solve underdetermined systems without any changes to the LP formulation.

The solution found will be the best in the sense of leaving the least tolerance that bounds the deviation of model outputs from targeted output values in the available training data. An underdetermined system always allows multiple solutions, and selecting one of them algorithmically should match some common sense criterion, as it does here. So a basic level of feature selection, down to n features, happens by the nature of the solution method. Further pruning of the features, to find the most important ones, is possible as follows. Consider again the condition for solution $\left|\breve{A}x-\breve{b}\right|+rad(A)\left|x\right|\le rad(b)$

If we force a component of x to zero, the first term changes at a rate proportional to coefficients in the respective column of $\breve{A}$, while the second term improves for minimization purposes at a rate proportional to the same column in *rad*(*A*). The improvement dominates in all equations for component 1, for example, if the first column in *rad*(*A*) dominates the first column of *A*: $rad(A)\hat{e}_1\ge\left|A\right|\hat{e}_1$

Then the optimal solution must have $x^*_1=x^*\cdot\hat{e}_1=0$. The condition given is sufficient, not necessary: components get eliminated also because of competition with other variables, and because it is the maximal component(s) in *rad*(*b*) that need to be reduced while the other inequalities are not similarly binding. But the inequality above proves beyond doubt that any component will be forced to zero by increasing *rad*(*A*) sufficiently, and doing this gradually is a way to leave less and less non–zero components in the optimal solution. By this means, we can in an orderly fashion pursue sparse solutions, by using in the LP upper and lower bounds that fake excess (for $t>1$) inaccuracy:

$$\left[A_{\min fake,}A_{\max fake}\right]\leftarrow\left[\breve{A}\pm t*rad(A)\right],\ \ with\ \ t\ge 0.$$

Now t is an extra parameter by which the sparsity of the solution is controlled, with larger values inducing higher sparsity, so there are less non–zero parameters in the model and it uses fewer inputs. Once the feature selection has been performed in this manner, for example, picking 3 features, the model parameters for these features should be re-determined with the real ($t=1$) inaccuracy. The trick with excess inaccuracy helps eliminate "inaccuracy sensitive" features, but also affects the parameters identified for the remaining ones, so these intermediate parameter values are not optimal for the true data inaccuracy.

Whether the sparsity of solutions comes naturally from the LP, or is forced further by requiring robustness against some excess level of inaccuracy, in MMV we face the problem of joint sparsity: the non–zero components of parameter vectors should coincide by indexed location, while their actual numeric values are not coupled.

In an LP the joint variable selection can be handled with a simple "large *M*" trick, which we illustrate with the simultaneous solution of equations

$$\left[A\right]x=b \text{ and } \left[A\right]y=c.$$

The LP performing joint variable selection is:
Minimize s

$$x_+\ge 0, x_-\ge 0, y_+\ge 0, y_-\ge 0, s>0$$

$$A_{max}x_+-A_{min}x_-\le b+se$$

$$A_{min}x_+-A_{max}x_-\ge b-se$$

$$A_{max}y_+-A_{min}y_-\le c+se$$

$$A_{min}y_+-A_{max}y_-\ge c-se$$

$$y_++y_-\le M(x_++x_-)$$

The last inequality performs the coupling with a large constant *M*, for example $M=10^5$. It can be written as $\left|y\right|\le M\left|x\right|$, which forces y to have zeroes where x does, and if x is already "too sparse" then y-values can be allowed with an in comparison small perturbation to the corresponding component of x. The optimal solution for x remains $x^*=x_+-x_-$, and y is similar.

The level of desired sparsity can again be enforced by adjusting excess inaccuracy with parameter t, and once the jointly used features, to predict both b and c, have been selected, their model parameters should be re-solved with the real level of inaccuracy at ($t=1$).

For an MMV problem with m simultaneous outputs, the number of unknowns is $2*m*p+1$, where p is the number of features available. This equals twice the number of elements in the X matrix, because these are split to positive and negative parts, plus one for the tolerance parameter s.

An alternative approach to force jointly sparse solutions, in a linear program, would be to introduce binary variables: 0/1 indicators of whether a feature is used or not. However, such binary variables make numerical LP solutions much harder to compute, requiring on top of basic Simplex algorithm also branch-and-bound –type algorithms. In comparison, the above "large *M* trick" is computationally efficient, as all variables remain continuous and a single run of Simplex suffices. Too large *M*-values can lead to numerical instability, which is the reason for giving a rule of thumb to set its value.

In summary, we have developed a theory and the LP formulations that allow robust jointly sparse solutions of multivariable multiple linear regression problems with errors-in-variables, for the particular case of errors bound within deterministic intervals. The approach has not required advanced concepts; instead it has remained on a pedestrian level in its requirements.

3. Results

3.1 A numerical example illustrating key points

Consider the square linear system $b=Ax$ with $b=(0,1,2)^T$ and $A=\begin{pmatrix}1&0&1\\0&1&1\\0&0&3\end{pmatrix}$ which is clearly non-singular and invertible. In such a case, the statistical method Total Least Squares (TLS) will always simply give the solution of this linear system, $x = A^{-1}b$, regardless of how inaccurate each element in A is. The square sum of errors is at its minimum which is zero, as neither the matrix nor the output needs tuning to match each other. To the contrary, the solutions from our approach are affected by the inaccuracies, and changing the level of inaccuracy in general changes the solution. This is a desirable feature of our approach compared with TLS, because we don't want to let a very inaccurate measurement affect the predictions of our regression model.

Assume the first feature/column is uniformly inaccurate, the second accurate, and the third proportionally inaccurate, with radius matrix $rad(A) = \begin{pmatrix} 0.2 & 0 & 0.5 \\ 0.2 & 0 & 0.5 \\ 0.2 & 0 & 1.5 \end{pmatrix}$. The accurate solution is $x = \left(-\frac{2}{3}, \frac{1}{3}, \frac{2}{3}\right)^T$, while the most robust solution against the large inaccuracy given is $x = \left(0, 0, \frac{2}{3}\right)^T$. In this case, even the fully accurate second variable drops out, showing that sparsification takes place by competition between the feature variables. The solution was numerically computed within a spreadsheet, with the LP formulation in the preceding Section Theory.

The purpose of this numerical example was to illustrate how easily different types of inaccuracy can be combined, and to provide a simple test case for anyone who writes code to carry out similar computations. Simultaneously a dramatic difference to TLS, which ignores matrix uncertainty when an accurate solution is possible, was made concrete. Our approach will drop inaccurate features, and even fully accurate features, if it increases robustness of the solution. Also, the solution will change with the level of inaccuracy, and all inaccurate features can be eliminated with sufficient excess inaccuracy. It may seem counterintuitive that the least accurate third feature provided the most robust solution above, and the fully accurate feature turned out to be irrelevant. An MMV problem with the same matrix and its radius is created by requiring simultaneous solution targeting output $c = (0,1,0)^T$. The accurate solution is $y = (0,1,0)^T$. When solved for the interval matrix, with joint sparsity requirement, we get the same x as before, while now $y = (0,0,0)^T$. The reason is, that the x solution has output tolerance $s = 1$ relative to targeted output b, and forcing the middle component of y to zero does not exceed this tolerance relative to targeted output c. So the best jointly sparse solution does not perturb x at all, which would worsen the tolerance, instead it forces y to zero in components where x is zero.

It is recommended that readers implementing this in a spreadsheet should try out various values of the sparsity-controlling parameter t, with $t = 0$ giving the accurate solutions, while $t = 1$ gives the other solutions listed above, and intermediate values show how the components of accurate solution are gradually forced to zero.

The small MMV problem demonstrates how sharing the same output tolerance is a key part of the coupling between the problems for x and y. The large M trick only couples the sparsity pattern, while the rest of the coupling also affects the numeric values. Applications should be scaled so that the tolerance levels of different outputs are similar, before they are combined in an MMV problem and solved as above.

4. Discussion and Conclusions

The theory presented provides several original contributions. Compressed sensing is a relatively new field with highly active research, where many algorithms to find the sparsest solutions to an underdetermined linear problem have been developed. The only deterministic interval approaches within that field, known to this author, are Rosenbaum and Tsybakov (2008), and very closely similar work by Liu *et al.* (2010). The former develops a non-convex formulation related to the Oettli-Prager theorem, as presented in Polyak and Nazin (2004), for the special case of uniform interval inaccuracies. A convex problem suitable for LP solution is only obtained by restricting all solved parameter values to be non–negative. These assumptions severely restrict the practical applicability of these prior results. In the theory of optimization, convexity is close to the same as "numerically solvable", while non-convex often translates to "requiring excessive computation, if solvable at all". In this case, only allowing non–negative parameters made the problem convex, so the restrictive assumption was made for a compelling reason.

In contrast, our approach is convex to begin with, and sparsity is induced by requiring robustness of solution against matrix inaccuracy. Our solution becomes sparser as we allow more inaccuracy, while Rosenbaum and Tsybakov separately pursue the sparsest solution within a convex portion of the Oettli-Prager solution set.

The robust regression work of Xu *et al.* (2009) is closely similar to ours, but limits the matrix uncertainty to featurewise norm bounds for whole matrix columns, with the same norm applied to output error. They focus on the Euclidean norm, although with some generalizations to other norms. In contrast we give a straightforward explicit LP approach to handling arbitrary error interval bounds, measurement by measurement, while the sup-norm of output error is in fact a uniform elementwise tolerance. The equations that describe the solution set sometimes known as tolerance solutions $\left|\breve{A}x - \breve{b}\right| + rad(A)\left|x\right| \leq rad(b)$ were here derived effortlessly along the lines given by Mayer (2007). A comprehensive and quite elaborate treatment of interval linear systems, including a more excruciating derivation of the equations above, is available in the book by Fiedler *et al.* (2006). This book can point the interested reader to relevant references from the recent about 50 years. While our references are very recent, the foundations have been laid much earlier.

The sparsifying effect of the "regularization term" $rad(A)|x|$ proven here provides a generalization to the L1-norm regularization practiced in compressed sensing, and goes beyond the connection earlier noted by Xu *et al.* (2009). Our regularization term in essence is a weighted L1-norm, allowing different weights for each case. Prior compressed sensing literature has not introduced such complicated weighing, because it has not considered measurement-by-measurement interval inaccuracy.

Statisticians are well familiar with the Lasso that is extensively discussed by Hastie *et al.* (2009), where least squares fit is regularized with the L1-norm $\|x\|_1$ (the sum of elements in $|x|$) An analogous special case is recovered from our theory when [*A*] is uniformly inaccurate and $rad(A) = ee^T$ with some constant scaling. The *e*'s are again vectors of ones, here not necessarily of the same dimension. On application of the Lasso, the regularization coefficient is indeterminate, and it is typically tuned with cross-validation. In our approach the regularization is fully determined by the inaccuracy in measured predictive features. However, additional sparsity to save in measurement costs, not for optimality in fitting, can be sought with the *t*-parameter that here seemed an *ad hoc* construct for those not familiar with the Lasso.

We have now shown with a wide generality that pursuing robustness against errors-in-variables leads to sparse or parsimonious solutions. While parsimony has been widely viewed as a virtue promoting simplicity, beauty, or interpretability, and compliance with "Occam's razor", here it is not a purpose in itself but emerges from robustness that engineer's desire. As noted earlier, the right level of parsimony remains hard to decide, and this is where statisticians resort to AIC and BIC used in model comparisons. For us, optimal robustness decides that right level automatically. This apparently significant difference is due to different assumptions. Our approach could be described as belonging to "approximation theory", while the statistical approach is probabilistic with assumed distributions, and leads to the complications with AIC and BIC.

Jointly sparse solutions have been pursued in various ways by Cotter *et al.* (2005), Mishali and Eldar (2008), and Lee and Bresler (2010), as select highlights among many others. The "large *M*" trick provided here for linear programming solution of the MMV problem is original, making the widely accessible LP solvers a reasonable tool to deal with such problems at least in small scale. This is in contrast to greedy approximations that proliferate in compressed sensing algorithms. With large scale problems though, the greedy methods remain undisputed.

The reader may note that both commercial and freely downloadable LP solvers are available, those that plug into a spreadsheet as well as others, that are designed to handle thousands of variables. Very large problems should still be reduced in size by pre-filtering the features, based on prior knowledge that depends on the context.

On the positive side, the simple approach we have disclosed empowers any typical researcher with access to a spreadsheet to create models that are robust against inaccurate inputs, with quite complicated patterns of inaccuracy if so happens. The tolerance parameter *s* solved shows how robust the model is: if *s* is large in comparison to target vector *b*, please reject the model. Such simple and understandable diagnostic is indeed concrete in comparison to AIC and BIC. Further, one can choose to pursue sparsity, and even joint sparsity (MMV), beyond the basic level induced by robustness against the true noise level in model features, simply by exaggerating the inaccuracy in computations.

On the negative side, the approach does have its limitations. The worst case estimates with intervals tend to be pessimistic, and our robust LP approach minimizes the worst case tolerance. Adding *k* variables with the same interval inaccuracy multiplies the interval width with *k*, while for random independent variables it is the variances that add the same way and the standard deviation only grows as $\sqrt{k}$. The worst case is extremely unlikely when the disturbances are random, unbiased, and independent, and worst case tolerances can be made irrelevant by this fact. Probabilistic approaches including Total Least Squares can then be worth considering, but the assumptions of Gaussian unbiased independent errors should not be made lightly. However, if the number of retained features remains small (i.e., *k* above is small), our approach is very reasonable.

Based on the assumptions required by alternative approaches and their known behavior, the worst case estimates and our linear programming formulations are most appropriate when

- The "noisy" predictive features are naturally described by finite intervals.
- The number of retained features in the final model is relatively small.
- There are batch effects and other biases, not well described as random unbiased independent noise.
- Hard performance bounds are required of the fitted model, by its eventual users.

As for the assumption of interval inaccuracy in the noisy predictive features, it does naturally fit quantization, binning, rounding or truncation of numbers, and physical situations where an actuator or measurement device has hysteresis or slack within given bounds, or is calibrated to perform within an interval accuracy specification. It does not fit well inherently stochastic processes.

The method presented is currently the only one known to the author that can deal with an underdetermined system with errors-in-variables, and still find jointly sparse robust solutions for multiple targeted outputs. The ease of understanding the method and implementing it will hopefully contribute to a wide range of future applications.

Future publications expanding this work may include minimizing the L1 norm of fitting error, while here we used the sup-norm as in Chebyshev approximation. When the inputs are accurate L1 fitting is known as LAD, least absolute deviations. The sup-norm is appropriate when the target values are reliable, while L1 suits the case with outliers or spike errors in the predicted variable. Also robust polynomial regression with a single inaccurate input may in the future complement the current multiple multivariable regression results, specifically with an approach doable with LP. An approach for robust polynomial regression, with matrix linear fractional transforms and advanced optimization methods, is given by El Ghaoui and Lebret (1997).

References

Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. 2009. Robust Optimization, Princeton University Press, Princeton, U.S.A., pp. 19-25.

Cotter, S.F., Rao, B.D., Engan, K. and Kreutz-Delgado, K. 2005. Sparse solutions to linear inverse problems with multiple measurement vectors. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing. 53, 2477-2488.

El Ghaoui, L. and Lebret, H. 1997. Robust solutions to least-squares problems with uncertain data. SIAM Journal on Matrix Analysis and Applications. 18, 1035–1064.

Fiedler, M., Nedoma, J., Ramik, J., Rohn, J. and Zimmermann, K. 2006. Linear Optimization Problems with Inexact Data, Springer Science+Business Media, New York, U.S.A., pp. 62-63.

Hastie, T., Tibshirani, R. and Friedman, J. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer Science+Business Media, New York, U.S.A., pp. 57-94.

Lee, K. and Bresler, Y. 2010. Subspace-Augmented MUSIC for Joint Sparse Recovery. Proceedings of Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 Institute of Electrical and Electronics Engineers (IEEE), Jerusalem, Israel, October 4-7, 2010, 205-208.

Liu, Y., Wan, Q., Wen, F., Xu, J. and Peng, Y. 2010. Anti-measurement Matrix Uncertainty Sparse Signal Recovery for Compressive Sensing. Arxiv repository (http://arxiv.org/abs/1006.0054).

Mayer, G. 2007. On regular and singular interval systems. Journal of Computational and Applied Mathematics. 199, 220–228.

Mishali, M. and Eldar, Y. 2008. Reduce and Boost: Recovering Arbitrary Sets of Jointly Sparse Vectors. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing. 56, 4692-4702.

Moore, R. 1979. Methods and Applications of Interval analysis, SIAM, Philadelphia, U.S.A., pp. 87-88.

Polyak, B. and Nazin, S. 2004. Interval solutions for interval algebraic equations. Mathematics and Computers in Simulation. 66, 207-217.

Rosenbaum, M. and Tsybakov, A. 2008. Sparse recovery under matrix uncertainty. The Annals of Statistics. 38, 2620-2651.

Tarantola, A. 2005. Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, Philadelphia, U.S.A., pp. 81-230.

Wiesel, A., Eldar, Y. and Yeredor, A. 2008. Linear Regression With Gaussian Model Uncertainty: Algorithms and Bounds. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing. 56, 2194-2205.

Xu, H., Caramanis, C. and Mannor, S. 2010. Robust Regression and Lasso. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Information Theory. 56, 3561-3574.