Imputing Missing Data for Gene Expression Arrays

Trevor Hastie[*], Robert Tibshirani[†], Gavin Sherlock[¶], Michael Eisen[‡], Patrick Brown[§], David Botstein[¶]

September 9, 1999
Technical Report, Division of Biostatistics, Stanford University

Here we describe three different methods for imputation. The first is based on a reduced rank SVD of the expression matrix, the second is based on K-nearest neighbor averaging, and the third is based on repeated regressions. We demonstrate the techniques on the human tumor data and a subset of the yeast data.

1 Imputation using the SVD

The singular value decomposition offers an interesting and stable method for imputation of missing values in gene expression arrays. The basic paradigm is

- Learn a set of basis functions or *eigen-genes* from the complete data.
- Impute the missing cells for a gene by regressing its non-missing entries on the eigen-genes, and use the regression function to predict the expression values at the missing locations.

[*]Depts. of Statistics, and Health, Research & Policy, Sequoia Hall, Stanford Univ., CA 94305. hastie@stat.stanford.edu

[†]Depts. of Health, Research & Policy, and Statistics, Stanford Univ, tibs@stat.stanford.edu

[‡]Life Sciences Division, Lawrence Orlando Berkeley National Labs & Dept. of Molecular. and Cell Biology, University of California. Berk.; eisen@genome.stanford.edu;

[§]Department of Biochemistry, Stanford University;pbrown@cmgm.stanford.edu

[¶]Department of Genetics, Stanford University;botstein@genome.stanford.edu

The regression paradigm also makes clear that to achieve good predictions, the number of eigen-genes (predictors) should be quite a bit smaller than the number of non-missing observations.

1.1 SVD imputation using a *clean* training set

Let X be the $N \times p$ expression matrix. For the human tumor cancer data these numbers are 6830×64. Let X^c be the subset of complete genes (2069) and X^m the remainder with at least one missing value per row. Consider the truncated SVD of X^c:

$$\hat{X}^c_J = U_J D_J V_J^T \tag{1}$$

where D_J is a diagonal matrix containing the leading $J \leq p$ singular values of X^c, V_J and U_J the corresponding orthogonal matrices of J right and left singular vectors. This rank-J SVD can be characterized in several ways. One that suits our purpose is that it provides the best rank-J matrix approximation to X^c; i.e. it solves the problem

$$\min_{M \text{ rank } J} ||X^c - M||^2 \tag{2}$$

where $||\cdot||$ denotes the Frobenius (sum-of-squares) norm.

We now interpret the solution from a regression point of view. Let x be any row of X^c, and consider the least squares regression of the p values in x on the eigen-genes v_1, $v_2, \cdots, v_J$, each p vectors. This regression solves the least squares approximation problem

$$\min_{\beta} ||x - V_J\beta||^2 = \min_{\beta} \sum_{\ell=1}^{p} (x_\ell - \sum_{j=1}^{J} v_{\ell j}\beta_j)^2 \tag{3}$$

with solution $\hat{\beta} = (V_J^T V_J)^{-1} V_J^T x = V_J^T x$ (since V_J is orthogonal) and fitted values $\hat{x} = V_j\hat{\beta}$. Thus $X^c V_J = U_J D_J$ gives all the regression coefficients for all the rows, and $\hat{X}_c = U_j D_J V_J^T$ all the fitted values. So once the V_J are found, the SVD approximates each row of X^c by its fitted vector obtained by regression on V_J.

This also suggests that for a row x from X^m with some missing components, we can impute the missing values by a similar regression:

$$\min_{\beta} \sum_{\ell \text{ non-missing}} (x_\ell - \sum_{j=1}^{J} v_{\ell j}\beta_j)^2 \tag{4}$$

Let V_J^* be the *shortened* version of V_J, with the appropriate rows removed (corresponding to the missing elements of x). The solution to (4) is $\hat{\beta} = (V_J^{*T} V_J^*)^{-1} V_J^{*T} x^*$, and the predictions for the missing elements are $V_J^{(*)}\hat{\beta}$, where $V_J^{(*)}$ represents the complement in V_J to V_J^*. Note that the columns of V_j^* are no longer orthogonal.

There are no intercepts in the regressions (3) and (4). It is customary to center the data before computing the SVD. This amounts to subtracting the ith row-mean $m_i^c = 1/p \sum_{\ell=1}^{p} X_{i\ell}^c$ from each element in row i. Since the eigen-genes will each have mean zero, including an intercept in the regression in (3) is trivial: it is the mean of x. Since the columns of V_J^* no longer have mean zero, we have to explicitly include an intercept in the regressions in (4).

One needs to select an appropriate order J. We suggest a method based on simulation later in this report.

1.2 SVD imputation using all the data.

The approach described so far implies the availability of a reasonable set of complete genes, and the incomplete ones do not contribute to the SVD basis. This can be wasteful if many genes have missing entries, and not possible if every gene has missing entries. For the tumor data, about two thirds of the genes have missing entries, so this approach is feasible. We now describe another approach, using all the data (and explicitly including the intercepts).

Solve the following problem:

$$\min_{U_J, V_J, D_J} ||X - m1^T - U_J D_J V_J^T||^* \tag{5}$$

Here $||\cdot||^*$ is a squared matrix norm, but a special one. It sums the squares of all the elements, except ignores those entries where X has missing data. m is a vector of means, one element per row of X. If there were no missing entries, the solution is standard: m is the vector of row means of X, and U_J, V_J, and D_J are obtained from the rank-J SVD of the centered X. Once the rank-J solution is "found" to this problem, use it to fill in the missing values for X.

It is natural to use iterative methods to solve this problem.

1. Initially set the missing entries to the mean of the non-missing entries for each row, producing a complete matrix X^0. Set $i = 0$.

2. Compute the SVD solution to (5) for the complete matrix X^i, and produce X^{i+1} by replacing the missing values in X by the fitted values from this solution.

3. Set $i \leftarrow i+1$ and repeat step 2 until $||M^i - M^{i+1}||/||M^i||$ is below some threshold ϵ (10^{-6}), where M^i is the entire fitted matrix (plus intercept) at the ith stage.

In practice this algorithm converges quite rapidly, typically 5 or 6 iterations.

Now here are two interesting facts:

1. The solution to this (5) is a fixed point of the iterative algorithm outlined above. In other words, if we solve (5), fill in the missing values, and then compute the usual SVD of this "complete" matrix, we get back the solution to (5). This suggests that the iterative algorithm might converge to the solution to (5).

2. Suppose we take the eigen-genes obtained from the solution to (5), and impute the missing values using the regression approach in (4). The imputations are the same as those obtained from (5).

The proofs are easy. For claim 1, the postulated solution makes zero error at the imputed values, and is best (in a sum-of-squares sense) at every other value. Hence any other solution would increase the sum-of-squared errors.

For claim 2, consider (5) and fix V_J at the solution values. The squared matrix norm $||\cdot||^*$ is a sum of squared vector norms, one for each gene. Each one is a least squares regression problem, summing over the non-missing entries, as in (4). Hence (5) also solves each of these problems, and in fact the entries of M^∞ (the converged solution)

2 Nearest-neighbor imputation

One concern with the SVD method is that it does a lot of borrowing strength from the bulk of the data, and may not do well for unusual genes not well represented by the leading eigen-genes. At the other end of the global-local spectrum we find nearest-neighbor methods.

Here is a simple K nearest neighbor algorithm for imputing the missing values in x^*:

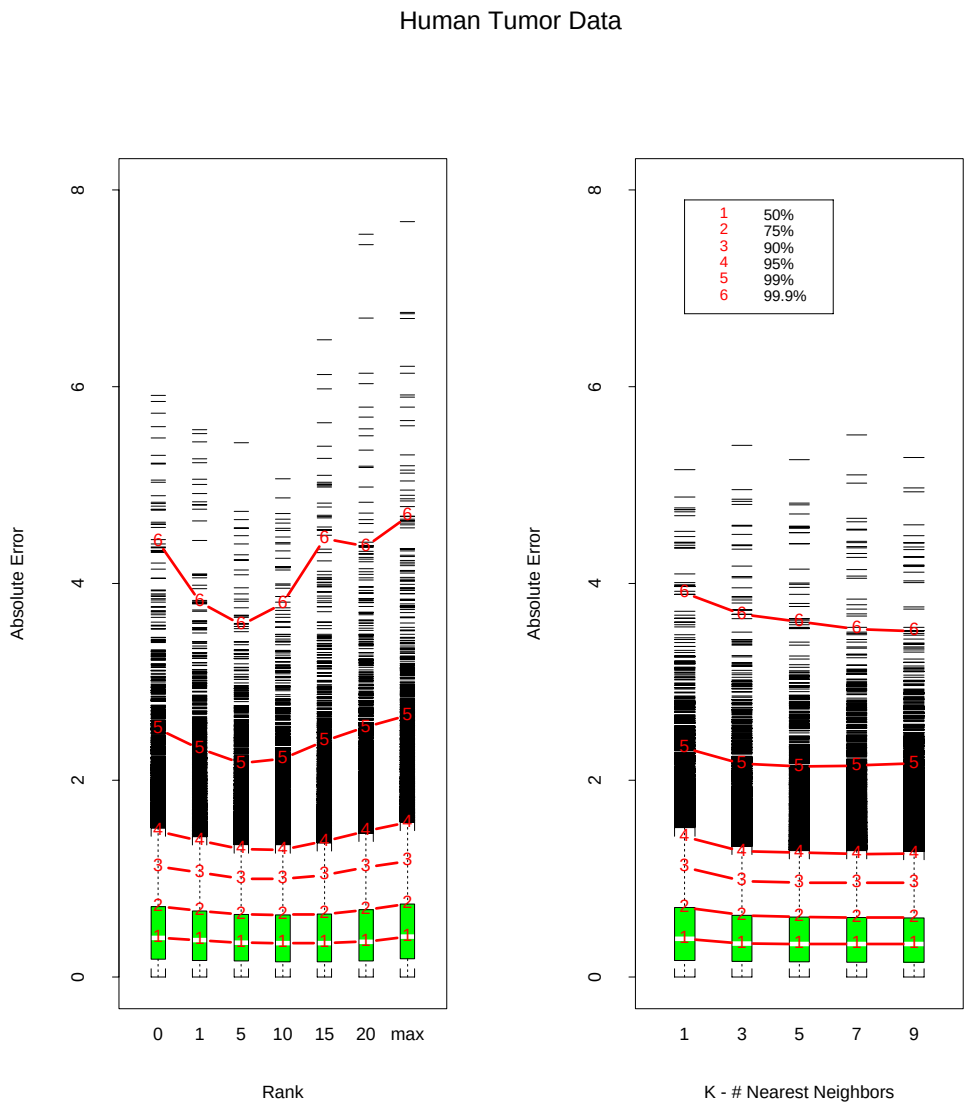

Figure 1: *Absolute errors of the SVD (left) and Knn imputations using the human tumor data. Missing data were imposed at random on the clean set* X^c. *In the left plot the missing entries were imputed using the SVD algorithm (5) for different ranks. Rank* 0 *corresponds to imputations using the mean, and* max *uses as many eigen-genes as possible. Shown are boxplots of the absolute errors, as well as selected quantiles. In the right plot, we see the corresponding picture for different size Knn imputations.*

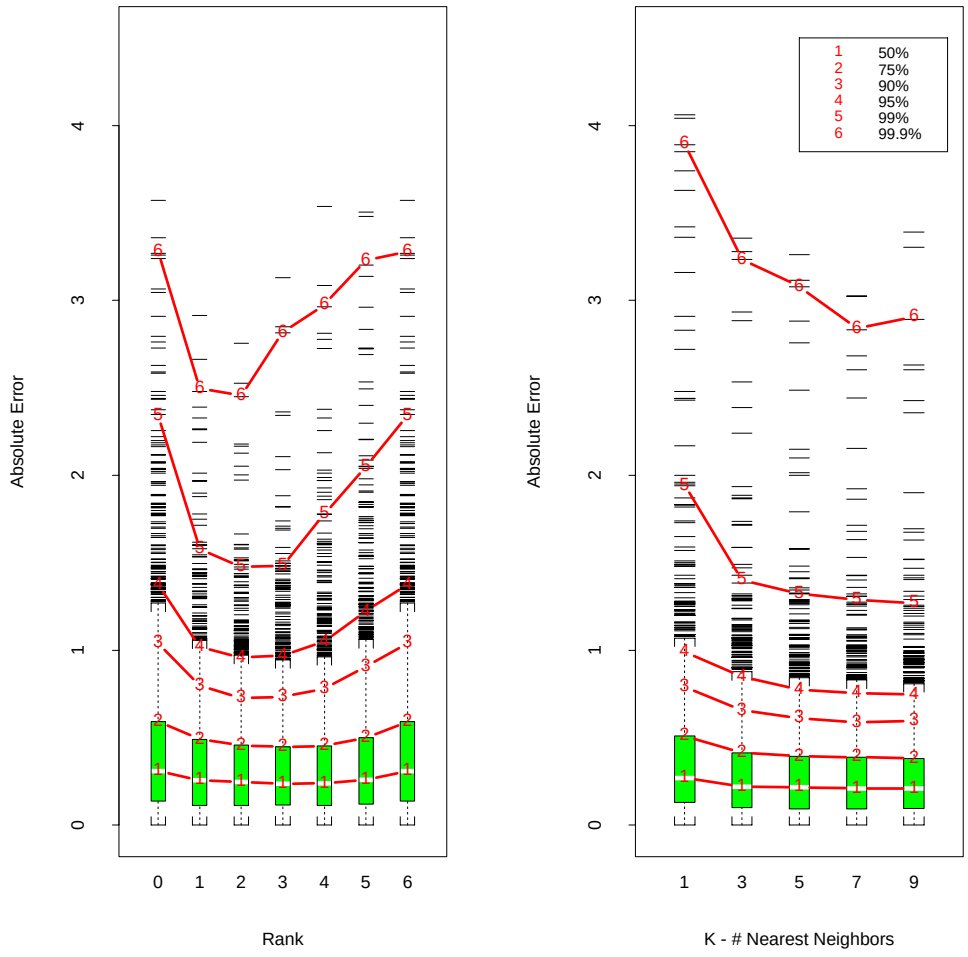

Figure 2: *Absolute errors of the SVD (left) and Knn imputations using a subset of the yeast data. The notation is as in figure 1.*

1. Compute the Euclidean distance between x^* and all the genes in X^c, using only those co-ordinates not missing in x^*. Identify the K closest.

2. Impute the missing coordinates of x^* by averaging the corresponding coordinates of the K closest.

Figure 5 shows that $K = 5 - 10$ is a good choice for the tumor data.

3 Imputation using regression

This technique is really intended for the case when the columns are *variables*, and the rows realizations of the variables. It is a standard EM approach for fitting multivariate gaussian means and covariances in the presence of missing data, and the imputed values come as a by-product.(Jerome Friedman, personal communication.) The idea is, for each j, to use regression of column j on every other column but j to impute the missing values in column j. In detail, for each column j in turn

1. Remove the rows of X which have missing values in column j.

2. Fit the regression of the *clean* column j on all the other columns (of this reduced X).

3. Use the coefficients from the regression to make predictions at the missing locations in column j.

Since their are missing observations in the other columns as well, this will not work as stated. Instead we use an iterative version (EM), where we always have imputed guesses (starting with the row averages) in each of the missing locations, and the imputations are updated as we proceed. The imputed values are used for the predictors in each regression, but not in the column designated as response.

This method seems to work very well, having a slight edge on both the Knn and SVD methods on the two arrays considered here. The iterations are rather slow to converge (as is typical of the EM).

The method could be generalized by using regression methods other than linear regression, such as regression trees, to perform the imputation for each column. In fact when CART is used, one can avoid iteration, because it can handle missing data in the predictors.

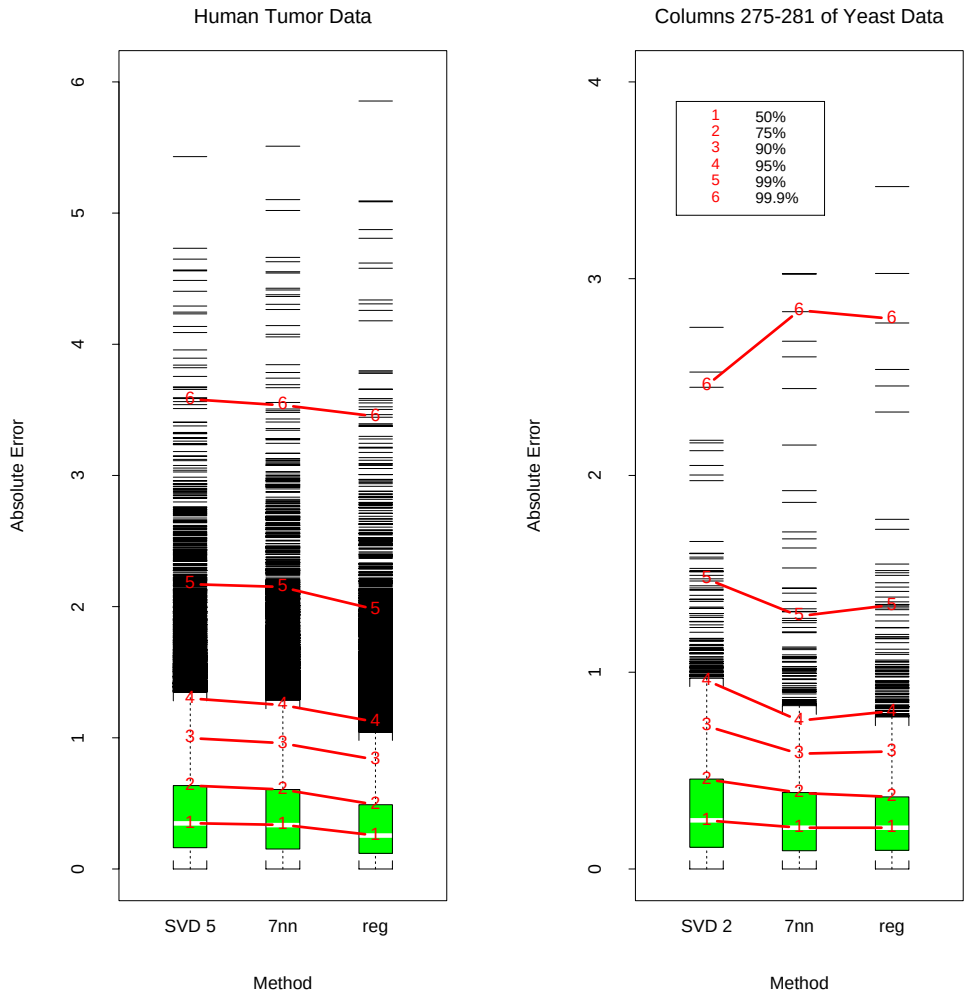

Figure 3: *Absolute errors of the best SVD and Knn compared with the regression method, for both data sets.*

4 Simulation to find J or K

For the tumor data, the expression values are not missing at random. More than 200 genes are missing in 10 or more positions, which is almost a zero probability event under the MAR assumption. In order to create realistic missing patterns, we randomly assign missing values to the elements in the 2069 rows of X^c by sampling 2069 rows from the 6830 rows of X, and use their missing locations. This lead to similar missing structure for the clean data set (70% missing rows, 3.3% missing values overall.)

For this contaminated version of X^c we impute the missing values for a range of values of J for the SVD, and a range of values of K for the Knn method. The boxplots in figure 5 are the absolute errors incurred, pooling 5 such random realizations.

The same strategy was used for the yeast data, which had 1.5% missing data, and 8% missing rows.

Figure 3 compares for both arrays, the best of the SVD and Knn with the regression technique. Although there is not much in the comparisons, it looks like the regression method has a slight edge.

5 Discussion

This is a working paper, and may change in the future. For a detailed comparison of these techniques, see Troyanskaya, Cantor, Sherlock, Brown, Hastie, Tibshirani, Botstein & Altman (2001).

References

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. & Altman, R. B. (2001), 'Missing value estimation methods for dna microarrays', *Bioinformatics* **17**(6), 520–525.