THE USE OF LINEAR ALGEBRA IN MODELING THE PROBABILITIES OF PREDICTED FUTURE OCCURRENCES

by

Gabrielle F.S. Boisramé

A thesis submitted in partial fulfillment of the requirements
for graduation with Honors in Mathematics.

Whitman College
2010

Certificate of Approval

This is to certify that the accompanying thesis by Gabrielle F.S. Boisramé has been accepted in partial fulfillment of the requirements for graduation with Honors in Mathematics.

Douglas Hundley, Ph.D.

Whitman College
May 12, 2010

ABSTRACT

THE USE OF LINEAR ALGEBRA IN MODELING THE PROBABILITIES OF PREDICTED FUTURE OCCURRENCES

Singular Value Decomposition (SVD) and similar methods can be used to factor matrices into subspaces which describe their behavior. In this paper we review the SVD and generalized singular value decomposition (GSVD) and some of their applications. We give particular attention to how these tools can be used to isolate important patterns in a dataset and provide predictions of future behavior of these patterns. A major focus of this project is the examination of a component resampling method described by Michael Dettinger which provides estimates of probability distributions for small sets of data [2]. We tested the results of using both the SVD and the GSVD for Dettinger's method. Similarly to Dettinger, we found that the method had a tendency to give probability distributions a Gaussian shape even when this did not seem to be represented in the original data. For some data sets, however, both using the SVD and GSVD provided what appear to be reasonable probability distributions. There was not a significant difference in how well original probability distributions were estimated when using Dettinger's original method or the modifications with the reduced SVD or the GSVD. Using Dettinger's method rather than a simple histogram always provided a higher resolution of information, and was sometimes capable of matching the shape of the original probability distributions more closely.

Gabrielle Boisramé
Whitman College
May 2010

Contents

List of Figures

Chapter 1

Introduction

1.1 Time Series and Matrix Decomposition

The accurate modeling and prediction of time series is becoming increasingly important in a range of applications, from meteorological forecasts to economic models. Along with the ability to predict a pattern comes the need to establish the reliability or accuracy of a prediction, preferably before the modelled event occurs. This paper addresses the issue of predicting the likelihood of an event from either a set of viable models for the event or a set of historical data. For example, figure 1.1 shows output from six different weather models which each predicted the maximum temperature in each day of a thirty day period. If all six models which contributed to the data in this graph are equally likely to be accurate, then whoever is analyzing this information is faced with the task of deciding how to describe what the maximum temperature is likely to be on each day given six different predictions. Common statistical tools such as means, medians, standard deviations, and histograms are certainly reasonable choices for describing temperature likelihood on any given day. This could lead to problems, however, if the data is heavily skewed to one side of the median, there are outliers, or there are too few data points to calculate a meaningful standard deviation or create a useful histogram. In such cases, simple statistics will not give a very complete picture of the actual probabilities. This paper addresses such problems by asking: How can we use tools such as *principal component analysis* and *independent component analysis* to help accurately describe the likelihood of a future event based on an ensemble of models for said event? To explore this question, we will test and expand upon a method proposed by Michael Dettinger, of the U.S. Geological Survey, for estimating probability distributions based on model ensembles [2]. This method provides a means of estimating probability distributions for time series described by small data sets. When there are only a small number of predictions for a future event, there may not be enough data points to provide a meaningful picture of the event's probability. Dettinger's method uses a principal component analysis (PCA) to divide the data into separate components, which are then redistributed in order to provide a large number of new "predictions" which follow the trends of the original but are now large enough in number to be able to provide more detailed information.

Such a method has the potential of being very useful for making policy decisions based on models for such things as weather or hydrology. It provides a means of taking the results of many different models into account without relying on any subjective decisions such as what data points count as outliers or which models appear most reliable.

Because of the nature of probabilities, it is impossible to know the underlying probability distribution for a predicted event with certainty. For example, it is always possible to get data points which do not provide a representative sample of a distribution, which would make reconstruction of the original probability distribution unlikely. At the least, however, Dettinger's method provides a means of describing a rough probability distribution from a small random sample. For small data sets on which traditional statistics cannot provide significant results, such an approximation can at least provide some framework for describing a probability on a more detailed level than simply giving means and standard deviations. From a purely mathematical standpoint, this method also provides an interesting example of an application of the SVD, GSVD, and other tools of principal component analysis.

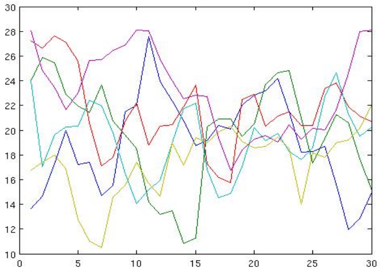

Figure 1.1: Six different models' predictions for maximum temperature over the same thirty day period in the same location.

1.2 Dettinger's Method

The basic idea behind Dettinger's *component resampling* method is that often there are multiple models for predicting a certain event, such as temperature or rainfall (these sets of models are called "forecast ensembles" by Dettinger)[2]. The models can either use different algorithms, have different input parameters, (i.e. climate models using different estimates for future atmospheric CO_2 concentrations) or both. Hopefully these models' predictions are fairly similar, though of course there will be some discrepancies, especially as the time from the initial conditions increases. Dettinger's goal is to provide a way of estimating what outcome within the range given by the forecast ensemble is most likely the mean (not the sample mean, which is easily computed, but the actual mean of the unknown underlying probability distribution), and especially to estimate what outcomes are the most likely.

Predicting the likelihood of an event requires a method of estimating its *probability distribution function* (PDF) from a set of data. A PDF is a measure of the probabilities of obtaining various outcomes for a given event (such as the roll of a die or the temperature at a given time), assigning a unique probability to each possible outcome in the domain. Describing a distribution involves some method of applying a regression to a set of data in order to find an algebraic expression for its PDF. If there are many models, then a fairly accurate PDF can be found by simply using a histogram at each time step and creating a mathematical expression to describe the histograms' shapes. Unfortunately, for small numbers of models this will not be very precise since the number of histogram bins it is possible to fill will be limited by the number of data points. To address this problem, Dettinger suggests decomposing the original forecasts into individual component vectors. Specific linear combinations of the component vectors will recreate the original data, but recombining the vectors randomly will fill in the gaps by creating a large number of forecasts which capture the same ranges and variations as the originals but are still distinct, such as those in figure 1.2. This large new set of data allows for more meaningful statistics because the sample size is so much larger. The data may be artificial, but it is still related to the original data, and is designed, essentially, to interpolate the information between the given data points.

Dettinger describes this method as analogous to filtering the original ensembles through many narrow, non-overlapping "frequency bands" to separate the components into bins according to their frequencies. Each separate forecast will have a different "power" in each frequency bin. Imagine that instead of dividing up groups of vectors the goal is to separate out the colors of many different beams of light by shining them through various polarized media. Light beam A might contain mostly red light, so it would have a relatively high power when looked at through something which only allowed red light through. Light beam B, however, might contain mostly blue light and only a little red, so it would have a higher power in the blue bin than the red. To make a new forecast, you take a power from each frequency bin, regardless of which model it originally came from, and put these all together (for the light analogy, a possible reconstruction would be a new beam of light created using the amount of red light from beam A and the amount of blue light from beam B). After doing this many times, powers which appear more often will appear in a proportionally larger number of the reconstructed models, and the trends from the original forecasts will be represented in this new, larger forecast ensemble.

This paper both analyzes results from using the component resampling method just described and extends upon it. The first goal is to test how accurately this method can reproduce PDFs from a limited sample. Next, we explore the use of different forms of principal component analysis and independent component analysis to possibly improve Dettinger's method. The main question we consider is whether removing noise from the data set will give more useful estimated PDFs, or if instead it would remove too much data to give a reliable prediction.

First, Dettinger's method is outlined in order to introduce the relationships between the topics which will be discussed in this paper. Further details on the justification for each step are given in later sections. We also discuss the background and applications of component resampling methods including *Singular Value Decomposition* and *Independent Component Analysis*.

For the equations in this paper, uppercase letters represent matrices, and bold lowercase letters represent vectors while other lowercase letters represent scalars. Generally, matrix notation will follow that in Golub and Van Loan [9], which is also an excellent reference for techniques described in this paper. Important terms and definitions are given in the appendices, and can also found in Lay, Linear Algebra and it's Applications [6] and Miller, John E. Freund's Statistics [7].

1.2.1 Dettinger's Component Resampling Method

Consider an ensemble of n different forecasts for the same set of m events. Each forecast $\mathbf{x}^j$, $1 \leq j \leq n$, is m time steps long with elements $\{x_1^j, x_2^j, \ldots, x_m^j\}$. For example, figure 1.1 shows a set of 6 different forecasts for the daily maximum temperature over 30 days; in this case $n = 6$ and $m = 30$. These forecasts can be compiled into an $m \times n$ matrix X, with a forecast in each column.

$$X_{m\times n} = \begin{bmatrix} x_1^1 & x_1^2 & \ldots & x_1^n \\ x_2^1 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_m^1 & x_m^2 & \ldots & x_m^n \end{bmatrix}$$

We want to factor X into

$$X = EP^T$$

with the columns of $E_{m\times m}$ being orthogonal vectors (the "frequency bins" or light polarizations in the above explanation). Each column of E is a vector $\mathbf{e}^k \in \mathbb{R}^m$, $1 \leq k \leq m$, written as $\begin{bmatrix} e_1^k & e_2^k & \ldots & e_m^k \end{bmatrix}^T$. Each vector has a corresponding coefficient in $P_{n\times m}$ (the "strengths" of each forecast in each bin). The k^{th} column of P is written $\mathbf{p}^k = \begin{bmatrix} p_1^k & p_2^k & \ldots & p_n^k \end{bmatrix}^T$. For example, the j^{th} element of the k^{th} coefficient vector, p_j^k, is the projection (strength) of the j^{th} ensemble member (column of X) on the k^{th} column of E. The goal is to resample the components in E according to the weights in P to get new reconstructed forecasts. We will call the set of reconstructed forecasts R.

These are the steps in Dettinger's component-resampling method. Justification for the more complex steps will be given in separate sections.

1. Calculate the mean values $\bar{x}_i$ for each time i across all forecasts:

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_i^j$$

2. Calculate the ensemble standard deviations s_i of the centered forecasts at each time i:

$$s_i = \left[\frac{1}{n} \sum_{j=1}^{n} (x_i^j - \bar{x}_i)^2 \right]^{1/2}$$

   Divide the centered forecast vectors at each time step by the corresponding standard deviation. This will create a *standardized* forecast ensemble, X' (mean of zero and standard deviation of one at each time step). Each entry in X' is given by

$$x_i'^j = \tfrac{(x_i^j - \bar{x}_i)}{s_i} \text{ for } 1 \leq j \leq n \text{ and } 1 \leq i \leq m.$$

   Subtracting the mean removes the mean shared by all the forecasts, and ensures that we reconstruct only the variations from this mean. Normalizing the standard deviation ensures that the variations between forecasts are treated in the same detail no matter how much the forecasts vary. After the ensemble is resampled we will reintroduce the mean and standard deviation.

3. Compute the $m \times m$ *Cross Correlation* Matrix $C = \frac{1}{n} X' X'^T$ which summarizes the *covariance* of each forecast with itself and each of the other forecasts at each time step. Defining c_r^s as the entry in the r^{th} row and s^{th} column of C, then each entry of C is found via

$$c_r^s = \tfrac{1}{n} \textstyle\sum_{j=1}^{n} x_r'^j x_s'^j \text{ for } 1 \leq r \leq m \text{ and } 1 \leq s \leq m.$$

4. - Construct the matrix E by setting each vector $\mathbf{e}^k$ equal to an eigenvector of C, i.e. $C\mathbf{e}^k = \lambda_k \mathbf{e}^k$. The corresponding eigenvalues, λ_k, represent the variance in X captured by the k^{th} vector of E.
   - Construct the coefficient matrix P by setting the columns of P equal to the projections of X' onto the vectors of E, so

$$\mathbf{p}^k = X'^T \mathbf{e}^k \text{ for } 1 \leq k \leq m$$

5. Construct additional "forecasts," $\mathbf{r}^l$, for l values ranging from 0 to whatever number of reconstructed "forecasts" is desired. Do this by recombining the columns from E and P randomly for each time step.

   Because of the way we created E and P, an exact reconstruction of each (standardized) data point would be found via $x_i^{'j} = \sum_{s=1}^m e_i^s p_j^s$, but the goal is to create distinct data sets. To do this, we will redistribute the individual components by randomly choosing the index j at each step in the equation, so the strengths of each eigenvector (or "frequency bin") in the constructed forecast is randomly chosen from all the strengths represented throughout the ensemble.

$$r_i^{'l} = \sum_{s=1}^{m} e_i^s p_{\text{random}(j)}^s$$

   Ideally there will be n^m distinct reconstructions possible, because there are n choices of coefficients p_j^s from each of the m columns of P. Since every coefficient vector does not contribute significantly to the variance of the reconstructions, however, the real number of significantly different reconstructions is usually smaller.

6. Rescale each new "forecast" to restore the original mean and scatter of the

ensemble. Essentially, undo the equation from step 2.

$$r_i^l(\text{rescaled}) = r_i^l s_i + \bar{x}_i$$

Now that we have a large number of “forecasts” they can be ranked and summarized in histograms to estimate PDFs. Fifty reconstructed “forecasts” from the example in figure 1.1 are shown in figure 1.2. Notice how they represent the general trends from the original 6 forecasts, despite all being different.

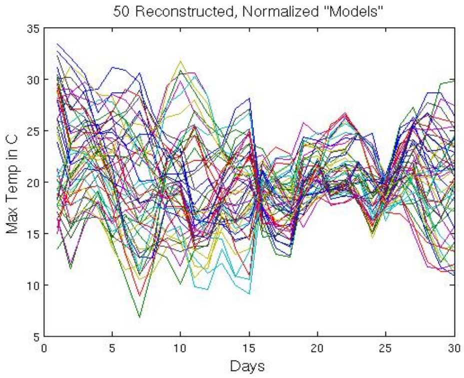

Figure 1.2: Result from steps 5 and 6 of Dettinger’s method.

To help clarify Dettinger’s method, here is a numerical example. The entries in X, though mostly random, have been chosen so that they could be imagined to represent three time series of four time steps each which share similar behavior, increasing until the third time step and then decreasing again at the end.

$$X = \begin{bmatrix} 1 & 3 & 2 \\ 5 & 4 & 5 \\ 10 & 12 & 14 \\ 7 & 7 & 8 \end{bmatrix}$$

1. Calculate the mean of each time step (row).

$$\text{Mean}(X)=\begin{bmatrix} 2.00 \\ 4.67 \\ 12.00 \\ 7.33 \end{bmatrix}.$$

2. Calculate the standard deviation of each row (we used the sample standard deviation, rather than the ensemble standard deviation, since our sample size is so small).

   $$\mathbf{s} = \begin{bmatrix} 1.00 \\ 0.58 \\ 2.00 \\ 0.58 \end{bmatrix}.$$

   Create the normalized forecast (notice that the mean is 0 and standard deviation is 1).

   $$X' = \begin{bmatrix} \frac{1-2}{1} & \frac{3-2}{1} & \frac{2-2}{1} \\ \frac{5-4.67}{.58} & \frac{4-4.67}{.58} & \frac{5-4.67}{.58} \\ \frac{10-12}{2} & \frac{12-12}{2} & \frac{14-12}{2} \\ \frac{7-7.33}{.58} & \frac{7-7.33}{.58} & \frac{8-7.33}{.58} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ .58 & -1.15 & .58 \\ -1 & 0 & 1 \\ -.58 & -.58 & 1.15 \end{bmatrix}$$

3. Compute $C = \frac{1}{n}X'X'^T$

$$C = \frac{1}{3}\begin{bmatrix} -1 & 1 & 0 \\ .58 & -1.15 & .58 \\ -1 & 0 & 1 \\ -.58 & -.58 & 1.15 \end{bmatrix}\begin{bmatrix} -1.00 & 0.58 & -1.00 & -0.58 \\ 1.00 & -1.15 & 0 & -0.58 \\ 0 & 0.58 & 1.00 & 1.15 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.67 & -0.58 & 0.33 & 0 \\ -0.58 & 0.67 & 0 & 0.33 \\ 0.33 & 0 & 0.67 & 0.58 \\ 0 & 0.33 & 0.58 & 0.67 \end{bmatrix}$$

4. Construct E and P^T:

   The eigenvalue/eigenvector pairs for C are:

$$\lambda_1 = 0, \mathbf{e}^1 = \begin{bmatrix} 0 \\ .35 \\ .61 \\ -.71 \end{bmatrix} \qquad \lambda_2 = 0, \mathbf{e}^2 = \begin{bmatrix} -.71 \\ -.61 \\ .35 \\ 0 \end{bmatrix}$$

$$\lambda_3 = 1.33, \mathbf{e}^3 = \begin{bmatrix} 0 \\ .35 \\ .61 \\ .71 \end{bmatrix} \qquad \lambda_4 = 1.33, \mathbf{e}^4 = \begin{bmatrix} -.71 \\ .61 \\ -.35 \\ 0 \end{bmatrix}$$

   so, one possible form of E (the columns can be in any order) is

$$E = \begin{bmatrix} 0 & -.71 & 0 & -.71 \\ .35 & -.61 & .35 & 0 \\ .61 & .35 & .61 & -.35 \\ -.71 & 0 & .71 & 0 \end{bmatrix}$$

Now that we have E, each entry of P is $\mathbf{p}^k = X'^T\mathbf{e}^k$, so

$$P = X'^T E = \begin{bmatrix} -1.00 & 0.58 & -1.00 & -0.58 \\ 1.00 & -1.15 & 0 & -0.58 \\ 0 & 0.58 & 1.00 & 1.15 \end{bmatrix} \begin{bmatrix} 0 & -.71 & 0 & -.71 \\ .35 & -.61 & .35 & 0 \\ .61 & .35 & .61 & -.35 \\ -.71 & 0 & .71 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.0048 & 0.0062 & -0.8188 & 1.06 \\ 0.0093 & -0.0085 & -0.8143 & -0.71 \\ -0.0035 & -0.0038 & 1.6295 & -0.35 \end{bmatrix}$$

5. Construct new forecasts, $\mathbf{r}^l$, where $r^l_i = \sum_{s=1}^m e^s_i p^s_{\text{random}(j)}$. For example:

$$r_1^{l1} = (0 \cdot .0093) + (-.71 \cdot 0.0062) + (0 \cdot 1.6295) + (-.71 \cdot -.35) = 0.2441$$

is one possible value for the first entry (first row) of any column representing a reconstructed forecast. The first scalar listed in each of the four products is just an entry from the first row of E, and the scalar by which they are multiplied is chosen from any row of P but always in the same numbered column as the entry in E. For this example, $(-.71) \cdot (-.35)$ is $e^4_1 p^4_3$.

6. Rescale each forecast. For our example, $r_1^{l1} = 0.2441$ would be rescaled according to the original mean and standard deviation of the first row of X.

$$r_1^{l1}(\text{rescaled}) = (0.2441 \cdot 1 + 2) = 2.2441$$

Notice that this fits within our original range for the first row of X.

1.2.2 A Closer Look at Resampling

Step 5 of Dettinger's method is the most important step in the process, as it is what resamples the data into new forecasts. It is also one of the more difficult steps to visualize. Below I have written out the factorization $X = EP^T$ with the bold column of E and row of P^T showing which vector multiplication gives which entry in X. This isn't new information, just a helpful visual.

$$\begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^n \\ x_2^1 & \mathbf{x_2^2} & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} = \begin{bmatrix} e_1^1 & e_1^2 & \dots & e_1^m \\ \mathbf{e_2^1} & \mathbf{e_2^2} & \dots & \mathbf{e_2^m} \\ \vdots & \vdots & \ddots & \vdots \\ e_m^1 & e_m^2 & \dots & e_m^m \end{bmatrix} \begin{bmatrix} p_1^1 & \mathbf{p_2^1} & \dots & p_n^1 \\ p_1^2 & \mathbf{p_2^2} & \dots & p_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^m & \mathbf{p_2^m} & \dots & p_n^m \end{bmatrix}$$

When we create the new forecast ensembles, we are essentially doing this for each new data entry: r_2^l is the product of this bold row $\begin{bmatrix} e_1^1 & e_1^2 & \dots & e_1^m \\ \mathbf{e_2^1} & \mathbf{e_2^2} & \dots & \mathbf{e_2^m} \\ \vdots & \vdots & \ddots & \vdots \\ e_m^1 & e_m^2 & \dots & e_m^m \end{bmatrix}$ and the column created by these bold entries $P^T = \begin{bmatrix} p_1^1 & \mathbf{p_2^1} & \dots & p_n^1 \\ \mathbf{p_1^2} & p_2^2 & \dots & p_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^m & p_2^m & \dots & \mathbf{p_n^m} \end{bmatrix}$ where the bold entries in P^T were chosen randomly, but only one from each row was chosen. Each reconstructed forecast, therefore, is given by $\boldsymbol{r}^l = E \begin{bmatrix} p_{\text{random}}^1 \\ p_{\text{random}}^2 \\ \vdots \\ p_{\text{random}}^m \end{bmatrix}$.

Clarifying Steps 3 and 4

Notice that in step 3 of Dettinger's method the cross correlation matrix is multiplied by $\frac{1}{n}$ rather than $\frac{1}{n-1}$ as in equation A.5, which is the more common definition of a cross correlation matrix. For some types of statistics this is an important distinction, but in our case we are only concerned with relative changes within our one data set. More specifically, Dettinger's method uses eigenvectors rather than raw data. Changing which scalar the matrix is multiplied by does not change its eigenvectors. For the purposes of Dettinger's method, using either version of the cross correlation matrix, or even using the *covariance matrix*, will yield the same final factorization of the matrix X.

For a better understanding of the information contained in the cross correlation matrix, C, let us go back to the numerical example given above. The diagonal entries of C are all the same, 0.67. This is because the matrix X' was standardized, so the variance within any one row is the same as the variance in any other row. Also, if we had divided the matrix by $n - 1 = 2$ instead of $n = 3$ the diagonal entries would have been 1, the standard deviation of our standardized rows. Each zero entry in C represents a pair of rows which are uncorrelated. For example, $c_{14} = 0$, so row 1 and row 3 are uncorrelated. As shown by the positive 0.33 in c_{31} and c_{13}, rows 3 and 1 are slightly positively correlated, meaning that as the entries in one row increase so do the entries of the other row, but not as quickly. Looking at the first and third rows of X', this relationship makes sense. The first row may not be continuously increasing, but on average it increases from left to right and so does the third row. Remember that each row in this case represents a time step in our time series, and the columns are different forecasts.

As an illustration of the eigenvectors which are used in Dettinger's method (step 4), figure 1.3 shows data from six different forecasts for temperature over a 21 day period. Figure 1.4 shows each column of E for this data (after being re-scaled).

Black stars in both graphs show the mean at each time step for the data shown in figure 1.3. Each data point in figure 1.3 is some linear combination of the data points in figure 1.4.

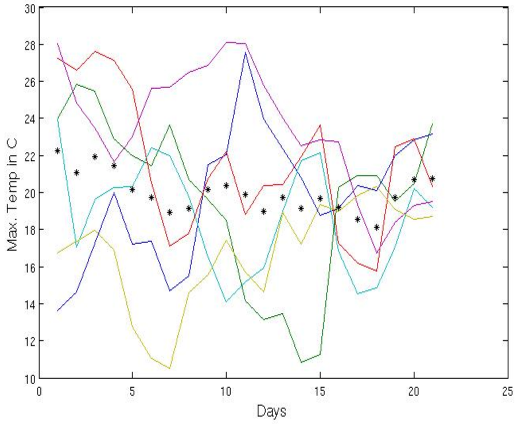

Figure 1.3: Six different forecasts for maximum daily temperature in the same location. Black stars show the ensemble mean.

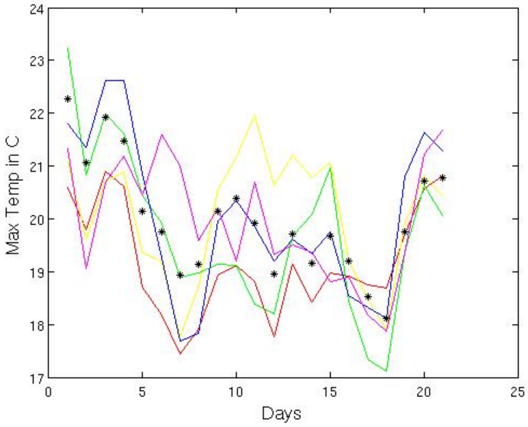

Figure 1.4: Re-scaled eigenvectors of data in figure 1.3. 1st Eigenvector:red, 2nd:yellow, 3rd:green, 4th:blue, 5th:magenta, 6th:black. Black stars show the original ensemble mean.

Chapter 2

Methods

2.1 Singular Value Decomposition

In the component resampling method just described, Dettinger chose to assign equal probabilities to all of the eigenvectors when resampling them to form new time series. A common practice with principal component analysis, however, is not to include those basis vectors which represent the noise in a data set. This paper examines the results of using Dettinger's method when some of the noise is removed, rather than using all of the basis vectors for the resampling. This alteration of Dettinger's method might create more precise PDFs, but it could also lead to an oversimplification of the data set and cause inaccuracies.

There are several recognized techniques for determining which of the cross correlation matrix' eigenvectors are the most important when it comes to describing overall behavior, and which only contribute to noise. The first such technique this paper will use is the *Singular Value Decomposition* (SVD).

The SVD is a method of factoring a matrix $X_{m\times n}$ into a product of matrices

$$X_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \tag{2.1}$$

where $U_{m\times m}$ and $V_{n\times n}$ are made up of the the orthonormal eigenvectors of XX^T and X^TX, respectively. The matrix $\Sigma_{m\times n}$ is a diagonal matrix with the square root of the eigenvalues of XX^T (and, consequently, of X^TX) in order from largest to smallest as its diagonal entries. If there are too few eigenvalues, then 0s are used to fill in the spaces and make Σ have the necessary dimensions. Symbolically, this gives:

$$\sigma_i = \sqrt{\lambda_i}$$

where λ_i is an eigenvalue of XX^T for any $X_{m\times n}$ and $\lambda_i > \lambda_{i+1}$.

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots \\ 0 & \sigma_2 & \cdots & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_k & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

To find the SVD of an $m \times n$ matrix X:

- Calculate XX^T and X^TX
- find the eigenvalues for either one
- find the eigenvectors for XX^T and X^TX, normalize all of them.
- Put these in matrix form: $U\Sigma V^T$. If there are not enough orthogonal eigenvectors to make U be $m \times m$ or V be $n \times n$, then construct additional normalized vectors which are orthogonal to the span of the eigenvectors. Pad any empty diagonal entries of Σ with zeros.

If the rank of X is k, the first k columns of U and V are k-dimensional bases for the columnspace and rowspace of X, respectively. The first j columns, where $j < k$, form a subspace which provides a lower-dimensional basis for the *best* possible approximation of X using j dimensions (see definition of "best" in Linear Algebra appendix). This method is often used to condense data into smaller matrices without losing much accuracy. It can also be used to determine important patterns in a set of data.

2.1.1 Proving the SVD gives the Best Basis

Assume that we are given a set of n data points $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ with each data point in $\mathbb{R}^m$ (Each $\boldsymbol{x}_i$ is $m \times 1$), where X is the $m \times n$ matrix with a data vector in each column. Given any orthonormal basis of $\mathbb{R}^m$, $\Psi_{m\times m} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_m]$ (Each $\boldsymbol{\psi}_i$ is also $m \times 1$) we can expand each vector $\boldsymbol{x}_i$ in terms of this basis:

$$\boldsymbol{x}_i = \sum_{k=1}^{m} \alpha_{k,i} \boldsymbol{\psi}_k = \Psi \boldsymbol{\alpha}_i = \Psi \left[\boldsymbol{x}_i\right]_\Psi \tag{2.2}$$

where $\alpha_{k,i}$ is a scalar entry in the vector $\boldsymbol{\alpha}_i = [\boldsymbol{x}_i]_\Psi$.

Since Ψ is orthonormal, the coordinates of $\boldsymbol{x}_i$ can be found via

$$\alpha_{k,i} = \boldsymbol{\psi}_k^T \boldsymbol{x}_i = \boldsymbol{x}_i^T \boldsymbol{\psi}_k. \tag{2.3}$$

In other words,

$$\boldsymbol{\alpha}_i = \Psi^T \boldsymbol{x}_i \tag{2.4}$$

Our goal in taking the SVD is to find the best way to reconstruct X from a set of basis vectors even if the number of basis vectors is smaller than the rank of X, so we need a way to express the error in a reconstruction. We will do this using the sum of the squared error: $\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_{approx(i)}\|^2$. The following steps show a way to decompose the matrix into a sum of norms, and then use this to calculate the error.

First, using equation 2.2 and the definition of a vector norm we find that

$$\begin{aligned} \|\boldsymbol{x}_i\| &= \sqrt{x_{1,i}^2 + x_{2,i}^2 + ... + x_{m,i}^2} && (2.5) \\ \|\boldsymbol{x}_i\|^2 &= x_{1,i}^2 + x_{2,i}^2 + ... + x_{m,i}^2 = \boldsymbol{x}_i^T\boldsymbol{x}_i && (2.6) \\ &= (\Psi\boldsymbol{\alpha_i})^T(\Psi\boldsymbol{\alpha_i}) = \boldsymbol{\alpha}_i^T\Psi^T\Psi\boldsymbol{\alpha}_i && (2.7) \\ &= \boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_i. && (2.8) \end{aligned}$$

Equation 2.7 works out the way it does because Ψ is orthonormal so $\Psi^T\Psi$ is the identity matrix. Rewriting our α's in terms of $\boldsymbol{x}$ and $\boldsymbol{\psi}$, with reference to equation 2.3, gives us

$$\|\boldsymbol{x}_i\|^2 = \boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_i = \sum_{k=1}^{n} \alpha_{k,i}\alpha_{k,i} = \sum_{k=1}^{n}(\boldsymbol{\psi}_k^T\boldsymbol{x}_i)(\boldsymbol{x}_i^T\boldsymbol{\psi}_k) \quad (2.9)$$

Summing both sides over all $\boldsymbol{x}_i$, we get

$$\sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2 = \sum_{i=1}^{n}\sum_{k=1}^{n}(\boldsymbol{\psi}_k^T\boldsymbol{x}_i)(\boldsymbol{x}_i^T\boldsymbol{\psi}_k) \quad (2.10)$$

Using the distributive property and the fact that the $\boldsymbol{x}_i$s are unaffected by k, we get

$$\begin{aligned} \sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2 &= \sum_{k=1}^{n}\left(\boldsymbol{\psi}_k^T\sum_{i=1}^{n}(\boldsymbol{x}_i)(\boldsymbol{x}_i^T)\boldsymbol{\psi}_k\right) && (2.11) \\ &= \sum_{k=1}^{n}(\boldsymbol{\psi}_k^T XX^T\boldsymbol{\psi}_k) && (2.12) \end{aligned}$$

When this is separated into the first $d < n$ vectors and the remaining vectors, we get

$$\sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2 = \sum_{j=1}^{d}\boldsymbol{\psi}_j^T XX^T\boldsymbol{\psi}_j + \sum_{j=d+1}^{n}\boldsymbol{\psi}_j^T XX^T\boldsymbol{\psi}_j. \quad (2.13)$$

Recall that X is the $m \times n$ matrix with a data vector in each column, so the first term of the right hand side of equation 2.13 is an approximation of X using d basis vectors, so if d is less than the rank of X there will be some error between this approximation and the actual X. This error is given by the second term of equation 2.13, which shows the difference between this reconstruction and the original matrix. Finding the best basis, therefore, will involve minimizing this error. Recall that our original goal was to minimize $\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_{approx(i)}.\|^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_{error(i)}\|^2$. Since $\boldsymbol{x}_{approx(i)}$ is a projection of $\boldsymbol{x_i}$, then $(\boldsymbol{x}_{error(i)})$ is orthogonal to $\boldsymbol{x}_{approx(i)}$ and therefore $\|\boldsymbol{x}_{error(i)}\|^2 + \|\boldsymbol{x}_{approx(i)}\|^2 = \|\boldsymbol{x}_i\|^2$. This can be easily visualized in two dimensions as a right triangle with legs of length $\|\boldsymbol{x}_{error(i)}\|$ and $\|\boldsymbol{x}_{approx(i)}\|$ and hypotenuse of length $\|\boldsymbol{x}_i\|$. This leads us to the fact that equation 2.13 can be viewed as

$$\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_{approx(i)}\|^2 + \sum_{i=1}^{n} \|\boldsymbol{x}_{error(i)}\|^2$$

and we want $\|\boldsymbol{x}_{error(i)}\|^2$ to be minimized. This is equivalent to maximizing

$$\|\boldsymbol{x}_{approx(i)}\|^2 = \sum_{j=1}^{d} \boldsymbol{\psi}_j^T X X^T \boldsymbol{\psi}_j.$$

Let $\boldsymbol{\phi}_i$ (an $m \times 1$ vector) and λ_i represent the ith eigenvector and eigenvalue of the matrix XX^T, respectively, and Φ be the matrix containing the eigenvectors as columns and Λ be a diagonal matrix with each λ as its diagonal entries. Since XX^T is symmetric, its eigenvectors are orthonormal and therefore $\Phi^{-1} = \Phi^T$. Using the spectral theorem, this means that $XX^T = \Phi\Lambda\Phi^T$.

We can write each $\boldsymbol{\psi}_j$ in terms of its coordinates with respect to the eigenvectors of $C = XX^T$. Remember that the basis is orthogonal. Also, we will call $\boldsymbol{\beta}_j$ the $(m \times 1)$ vector of coordinates for $\boldsymbol{\psi}_j$ in terms of Φ. This gives the following relationships:

$$\begin{aligned}
\boldsymbol{\psi}_j &= \Phi(\Phi^T\boldsymbol{\psi}_j) = \Phi\boldsymbol{\beta}_j \qquad \text{for } \; i = 1..m & (2.14)\\
\boldsymbol{\beta}_j &= \Phi^T\boldsymbol{\psi}_j & (2.15)\\
\boldsymbol{\beta}_{ji} &= \boldsymbol{\phi}_i^T\boldsymbol{\psi}_j = \boldsymbol{\psi}_j^T\boldsymbol{\phi}_i. & (2.16)
\end{aligned}$$

Since C is symmetric, by the spectral theorem $C = \Phi\Lambda\Phi^T$ and equation 2.15 gives

$$\boldsymbol{\psi}_j^T C\boldsymbol{\psi}_j = \boldsymbol{\psi}_j^T\Phi\Lambda\Phi^T\boldsymbol{\psi}_j = \boldsymbol{\beta}_j^T\Lambda\boldsymbol{\beta}_j \tag{2.17}$$

Applying equation 2.17 to the reconstruction of the data using a d-dimensional basis gives

$$\begin{aligned}
\sum_{j=1}^{d}\boldsymbol{\psi}_j^T XX^T\boldsymbol{\psi}_j &= \sum_{j=1}^{d}\boldsymbol{\beta}_j^T\Lambda\boldsymbol{\beta}_j = \sum_{j=1}^{d}\lambda_1\boldsymbol{\beta}_{j1}^2 + ... + \lambda_m\boldsymbol{\beta}_{jm}^2 & (2.18)\\
&= \lambda_1\sum_{j=1}^{d}\boldsymbol{\beta}_{j1}^2 + ... + \lambda_m\sum_{j=1}^{d}\boldsymbol{\beta}_{jm}^2 & (2.19)
\end{aligned}$$

We have just re-written the term in equation 2.18 that we want to maximize.

Recall that the coefficients in equation 2.18 can be written as

$$\boldsymbol{\beta}_{ji} = \boldsymbol{\psi}_j^T\boldsymbol{\phi}_i, 1 < j < d, 1 < i < m$$

which is equivalent to saying that each coefficient vector is the coefficient vector of the projection of $\boldsymbol{\phi}_k$ onto the subspace spanned by the $\boldsymbol{\psi}$'s. The projection of a vector cannot be any longer than itself, and each vector is orthogonal so its norm is 1, and therefore

$$\sum_{j=1}^{d} \boldsymbol{\beta}_{jk}^2 = \|\mathrm{Proj}_{\Psi}\left(\boldsymbol{\phi}_k\right)\|^2 \leq 1$$

with equality iff $\boldsymbol{\phi}_k$ is in the span of the columns of Ψ.

Therefore, the maximum of equation 2.18 occurs when the coefficients are equal to 1, which occurs if Ψ is the same as Φ. In conclusion, the best d-dimensional subspace for approximating X is formed by the span of the first d eigenvectors of XX^T.

□

Recall that in the SVD we wrote X as $U\Sigma V^T$, where $U_{m\times m}$ and $V_{n\times n}$ were formed by the orthonormal eigenvectors of XX^T and X^TX, respectively, and $\Sigma_{m\times n}$ is a diagonal matrix with the square root of the eigenvalues of XX^T (and, consequently, of X^TX) from largest to smallest as the entries on the main diagonal. Since the eigenvectors of XX^T are the same as the entries in U, then the best d-dimensional basis for an approximation of n data points in $\mathbb{R}^m$, or the column space of X,is the first d columns of U. A similar proof shows that the first d columns of V form the best basis for approximating the row space of X. This paper will focus on the column space and the U matrix, since the time series data was originally introduced as being set up in columns.

2.1.2 Note on the SVD and Uniqueness

Although the vectors in the U and V matrices found by taking the SVD of a matrix X always give the best bases (in terms of minimizing error in reconstruction) for the column space and row space of X, there is not a unique solution. As an example, the SVD decomposition of $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ can be written as

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

or

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

The only difference is that one vector in V is written either as $\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$ or $\begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$. These vectors point opposite directions, but have the same slope (speaking in terms of Cartesian Coordinates) and projecting a vector onto one is the same as projecting it onto the other except for a sign difference. For the purposes of the SVD, both vectors describe the same subspace. As this example shows, getting a specific vector is not the point of the SVD; the point is to describe the subspace which best encapsulates the data in the original matrix.

2.1.3 Dettinger's Method and the SVD

Part of this paper's purpose is to analyze the results of modifying which components of the data are resampled in Dettinger's method. The SVD decomposes data matrices much the same way as Dettinger's method, using eigenvectors of the covariance matrix. In Dettinger's papers, the matrix $X_{m \times n}$ is decomposed into $X_{m \times n} = E_{m \times m} P^T_{m \times n}$. The columns of E are the eigenvectors of $\frac{1}{n} XX^T$. In the SVD, the matrix U provides a basis for columns of data. We found the columns of the matrix U by taking the normalized eigenvectors of XX^T and ordering them according to the size of their corresponding eigenvalues. Dettinger's method does not specify an order for the columns of E, so the ordering from the SVD would be just as correct to use as any other. The difference between $\frac{1}{n}XX^T$ and XX^T and the fact that U is normalized but E is not normalized is not important in terms of eigenvectors, because the only difference is a constant multiple, and the constant multiple of any eigenvector gives another eigenvector. For example, $X = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ has eigenvectors that are a constant multiple of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ for any values of a and b.

In summary: The vectors in Dettinger's matrix E are constant multiples of the columns of U from the SVD, so U is a possible choice for E.
If U is used for E, then ΣV^T can be used for P^T, as is demonstrated here:

We are given $P = X^T E$, so $P^T = (X^T E)^T = E^T X$. If $E = U$ and $P^T = \Sigma V^T$, then

$$P^T = E^T X \rightarrow \Sigma V^T = U^T X.$$

Since $X = U\Sigma V^T$ and the columns of U are orthonormal, this gives

$$\Sigma V^T = U^T U \Sigma V^T = \Sigma V^T$$

This may not be a very rigorous proof, but it shows that it works to set $E = U$ and $P^T = \Sigma V^T$.

2.1.4 SVD Example: Eigenfaces

In order to illustrate some uses of the SVD, let us look at the example of data compression with image files. This paper is not especially concerned with compressing data, but this is a good example of how the SVD combines data sets.

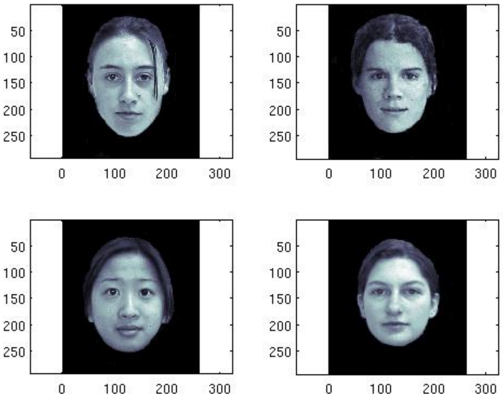

Figure 2.1: Four of the 30 faces used in the eigenfaces example. Source: Hundley [4].

Images such as those shown in figure 2.1 can be stored as vectors of numbers, each number representing the value from white to black of an individual pixel. Consider thirty of such photos, each one consisting of 77,028 pixels. Storing thirty of these pictures would take up a lot of computer memory. If we create a matrix with each vector of picture data as one column, then we can take the SVD of that matrix. The first few columns of U hold the most information about the faces, so in order to save memory space we can remove the higher-numbered columns of U. For example, let's take the first 15 columns of U. After getting the coordinates of a projection of the original matrix onto U, we can store most of the information as a $77,028 \times 15$ matrix $\hat{U}$ consisting of the first 15 columns of U as well as a 15×30 matrix containing the coordinates of each photo in terms of $\hat{U}$. This means the computer only has to store $15 \times 30 + 77,028 \times 15 = 1,155,870$ numbers instead of $77,028 \times 30 = 2,310,840$ numbers. The amount of storage space needed has been cut in half. Faces reconstructed from this compressed data storage are shown in figure 2.2. All of the data has not been retained, but the faces are still somewhat recognizable.

Dettinger's method can also be applied to these images. Figure 2.4 shows the result of using this method to combine the columns of U (some of which are shown

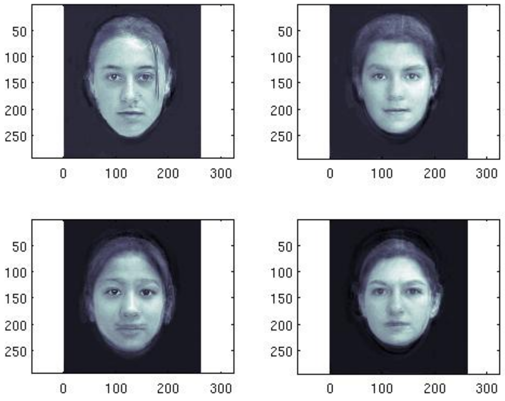

Figure 2.2: Same faces as in figure 2.1, reconstructed from the first 15 columns of U from the SVD.

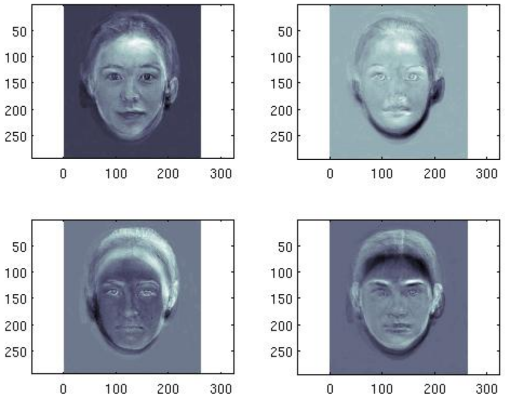

Figure 2.3: Images represented by the first four columns of U from the SVD.

in figure 2.3) using four different random choices of weights from the matrix P^T. Most likely, these faces will not match up with any of the original faces very well, but they still capture features represented in the original faces. This is what is happening with the time series data that Dettinger's method was originally designed to analyze: characteristic features of the original data are recombined to give new data that still fits within the original frame of reference. A reconstructed time series of temperature over a given year would be expected to have high temperatures in the summer and low temperatures in the winter, like the original time series it came

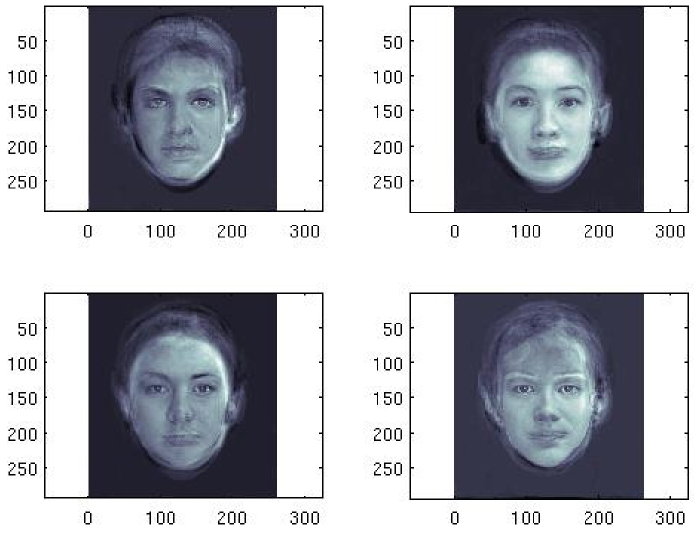

Figure 2.4: Examples of data from thirty faces resampled according to Dettinger's method.

from, just as the reconstructed faces in figure 2.4 still look like faces.

2.1.5 Reduced SVD and Eigenvectors

Now we will look into using the *reduced SVD* (RSVD) more carefully. By "reduced" we mean keeping d columns of the matrix U (or E, in Dettinger's notation) where d is less than the rank of the original data matrix X. According to the proof given earlier, the first columns of U in the SVD define the subspace which gives the closest approximation of the original data possible. The highest-numbered columns are generally assumed to be noise.

Looking at data from a toy data set (artificial data created simply as an example), we will be able to examine the results of using the reduced rather than the full SVD to create E and P^T in Dettinger's method.

Figure 2.7 shows the reconstructed forecasts (from Dettinger's method) using only part of the matrices E and P^T. For example, the points plotted in the second

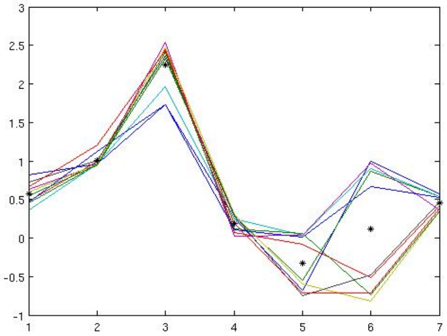

Figure 2.5: Ten different toy forecasts. Black stars show the ensemble mean.

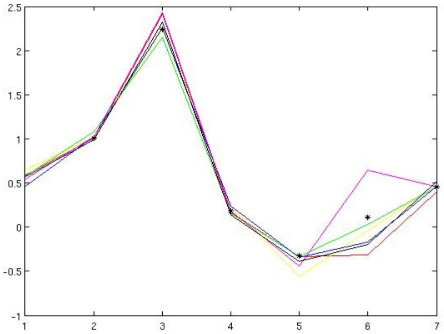

Figure 2.6: Eigenvectors of the data in figure 2.5, after re-scaling. 1st Eigenvector: red, 2nd:yellow, 3rd:green, 4th:blue, 5th:magenta, 6th:black. Black stars show the original ensemble mean. All of the lines shown in the graphs of figure 2.7, as well as the original data in figure 2.5, are linear combinations of these eigenvectors.

graph (counting from the upper left) are formed from multiplying the first two columns of E, $\hat{E}_{m\times 2}$, by vectors formed by choosing random entries in P^T as described in Dettinger's method, but only from the first two rows of P^T. The first few graphs don't seem to have very many reconstructions because there are fewer possible combinations of vectors from E and P. The algorithm ran 5,000 times, but most of the reconstructions were repeated and these repetitions just look like one line because they are lain exactly on top of one another. In fact, for the first column there is only one possible set of n (in this case n=10) reconstructions since the matrix of reconstructions is $L_{m\times n} = \hat{E}_{m\times 1}\hat{P}^T_{1\times n}$. There is only one possible choice from the columns of P^T for each $r_i^l = \sum_{s=1}^{m} e_i^s p^s_{\text{random}(j)}$ (recall that s is the number of reconstructions, i gives the time step, and l is the indexed number of the reconstructed vector) so there are only ten possible vectors for L. Notice that in the first graph of figure 2.7 the reconstructions follow the areas of greatest density in figure 2.5. By the next graph another layer of complexity is added. Also, notice that the second to last graph, which uses all but the last columns of E and P, is very similar to the final graph, which uses all of E. The graphs constructed using more vectors of U provide a greater number of distinct data points. The question is, however, whether all of these data points are truly necessary of if they are simply accentuating noise from the original predictions. This will be explored in more detail in the results section.

For the rest of this paper, when we say we used Dettinger's method with a reduced SVD (RSVD) we mean that E and P^T are given by some number of columns in U and ΣV^T, respectively.

2.2 Non Orthogonal Bases

The SVD often does not separate out noise very well when a data set already has some clear basis vectors which are not orthogonal. This is because the basis vectors from the SVD find patterns in the data by pointing in the direction of greatest variance, but the vectors must also be orthogonal. If there are several separate directions of high variance which are not orthogonal to each other, then the SVD cannot capture all of them.

We have already discussed using a reduced SVD to remove noise from the reconstructions in Dettinger's method. Later we will also devise a method of creating a factorization for Dettinger's method which accounts for the possibility of data with non-orthogonal bases which might not have its noise removed well by an

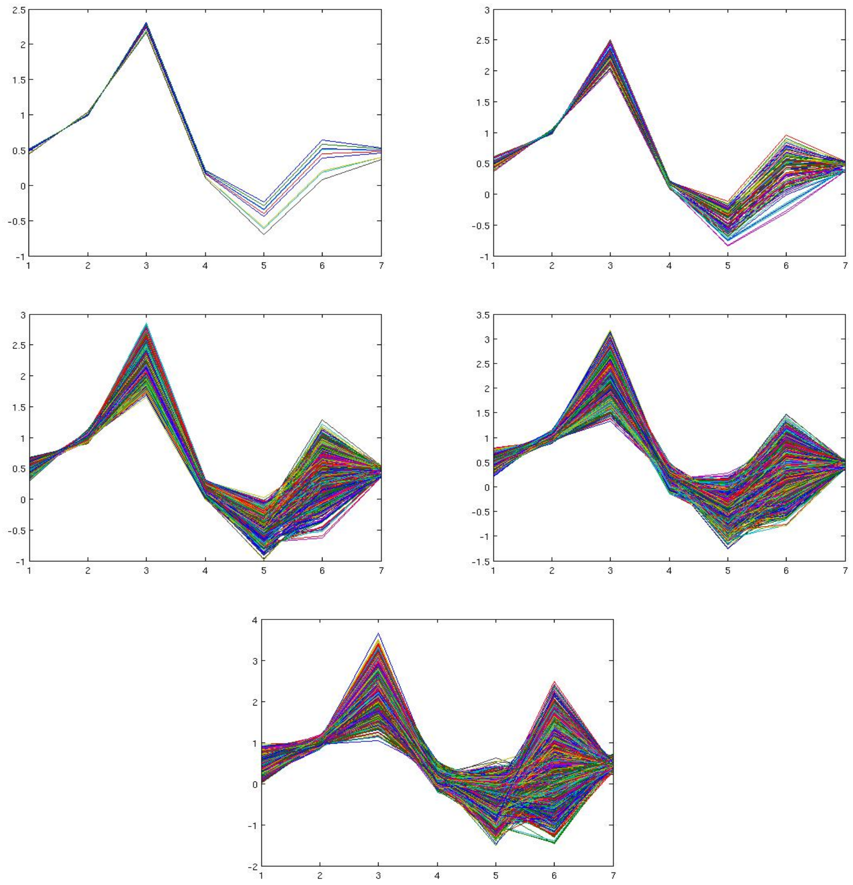

Figure 2.7: Starting in the top left, these graphs show reconstructions from: the first column of U, first two columns, first three, first four, and the full matrix U. These columns are the eigenvectors shown in figure 2.6.

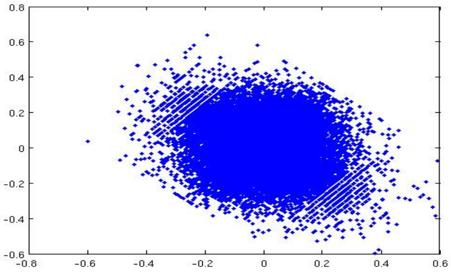

Figure 2.8: Data with a naturally occurring non-orthogonal basis.

SVD decomposition.

2.2.1 Independent Component Analysis

As an example of a naturally occurring non-orthogonal basis, imagine there are two microphones in a room and two different people are talking, so each microphone captures sounds coming from both speakers. The sound data from these microphones might look something like figure 2.8, which shows two vectors of "scrambled" sound data plotted relative to each other (the $x-$coordinates come from the vector of sounds recorded by the first microphone and the $y-$coordinates come from the second microphone). Rather than being scattered completely randomly, many of the points seem to lign up along slanted lines, or axes. These axes, however, are not necessarily orthogonal. Listening to the sound produced by each vector separately would sound like a mix of the two people talking, but ideally we would like to have only one clear voice represented in each data vector. Such a data set must be orthogonalized before we can compute a basis which will allow us to separate the two signals.

One method of orthogonalizing matrices to separate out clean signals is is called ICA, or *Independent Component Analysis*.

The basic idea behind the ICA is to decompose the data matrix, X, into the form

$$X_{m\times n} = S_{m\times n}A_{n\times n}.$$

Matrix S contains the columns of unscrambled data, which are the independent components of the ICA (n=2 columns for our sound data example). Matrix A is the "mixing matrix" which turns the clean data in the columns of S into the scrambled data in the columns of X [4].

Looking at A in terms of its SVD, $A = U\Sigma V^T$, we can get some idea of how this works. Since $X = SA$, we can also say that $X = SU\Sigma V^T$. Since the columns of U and V are orthonormal, multiplication by these matrices does not change the scale of the matrix S or the angles of its component vectors, they merely rotate it. Since Σ is a diagonal matrix, multiplication by Σ does not rotate the vectors, but it does scale them according to the entries in Σ. In other words, multiplying S by A is equivalent to multiplying S by U (a rotation), then multiplying the result by Σ (a scaling), and then multiplying that by V^T (another rotation). To undo this mixing, we take our data with its skewed axes, as in figure 2.8, then rotate it, scale it, and rotate it again to move our bases so that they create an orthogonal coordinate system for our data. The data can then be separated by looking at points along only one axis at a time.

There are many possible candidates for the matrices A and S to choose from, so we specify a set of criteria for our basis:

1. The resulting "clean" data should be smooth rather than change erratically with time. Looking at each vector as a time series, a small change in t should only result in a small change in y.

2. We want to separate out the data points which don't belong together, so the resulting data vectors should be statistically independent, not just linearly independent (see definitions in the Important Definitions and Equations section under "Linear Algebra").

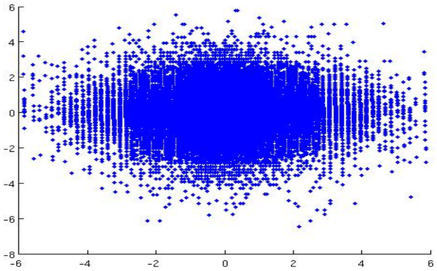

Figure 2.9: The Columns of S, the principal components referred to in the ICA.

The details of ICA will not be discussed here. For now, it is only necessary to understand that ICA takes scrambled data with a non-orthogonal basis (as in figure 2.8) and creates orthogonal basis vectors for the independent components within the data (as in figure 2.9). In this way, it can create a basis which more accurately represents the separate signals, or patterns, than simply using the SVD.

2.2.2 GSVD

A factorization for the ICA can be found using the *Generalized SVD* (GSVD), which is described in more detail in [9] and [10]. Under the right circumstances, this method will specifically separate the noise of a data set from the clean signals. In general, the GSVD of matrices $A_{m\times n}$ and $B_{p\times n}$ is a factorization of the form

$$A = UCZ^T \qquad \text{and} \qquad B = VSZ^T$$

where $C^TC + S^TS = I$.

To find the ICA factorization of a data matrix $X_{m\times n}$, we take the GSVD of X and its *difference matrix* dX. The difference matrix is an approximation of the derivative of matrix X with a time step of $\delta t = 1$. It is an $(m-1)\times n$ matrix of the differences between all adjacent data points.

$$dX = X(2:m,:) - X(1:m-1,:)$$

which can also be written as

$$dX = \begin{bmatrix} x_{21} - x_{11} & x_{22} - x_{12} \\ \vdots & \vdots \\ x_{m1} - x_{(m-1)1} & x_{m2} - x_{(m-1)2} \end{bmatrix}.$$

The reason for using the difference matrix stems from our criterion that the data should be smooth. The independent components are given by the columns of matrix U, where $X = UCZ^T$.

An example which illustrates the difference between the SVD and ICA (as produced using the GSVD) is the noisy circle. Figure 2.11 shows the x, y, and z components of a 3 dimensional circle with random noise added (figure 2.10). Figure 2.12 shows the result of trying to separate out the noise using the SVD. The noise has just been redistributed throughout the subspace vectors. Figure 2.13 shows the results using the GSVD. Because the GSVD orthogonalizes non-orthogonal subspaces, it is able to separate the noise from the clean signals.

In section 3.1.2 we will examine a version of Dettinger's method which uses the GSVD to create the component vector matrix E and the coefficients matrix P^T. Since we have shown that for some data sets the SVD does not separate out noise as effectively as the GSVD, we will test whether data analyzed by Dettinger's method is also represented better using the GSVD. To do this, we will use U from the GSVD for Dettinger's E matrix, and CB^T for his P^T. As with the reduced SVD, we can remove columns of E and P^T to remove the noise.

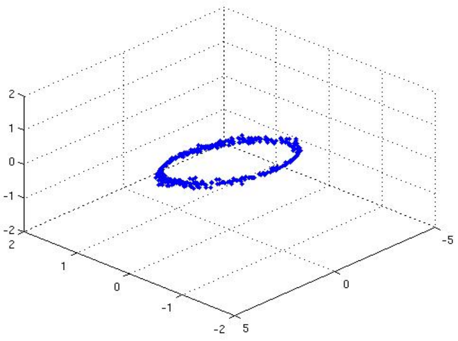

Figure 2.10: The three dimensional noisy circle referred to in figures 2.11 through 2.13.

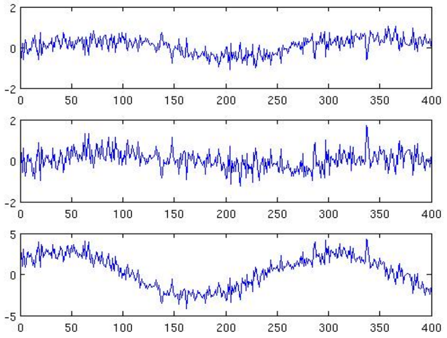

Figure 2.11: x,y,and z coordinates for a noisy circle.

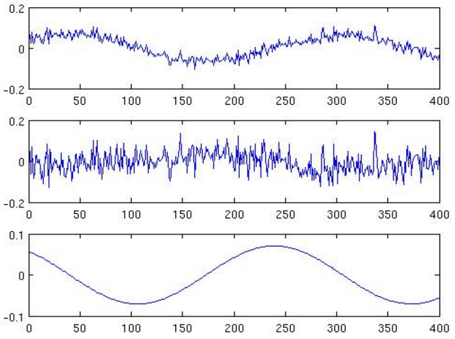

Figure 2.12: The columns of U from the SVD. The noise is distributed between the first two columns, because they are capturing the greatest amount of variance possible. The SVD cannot provide a clean signal for the circle.

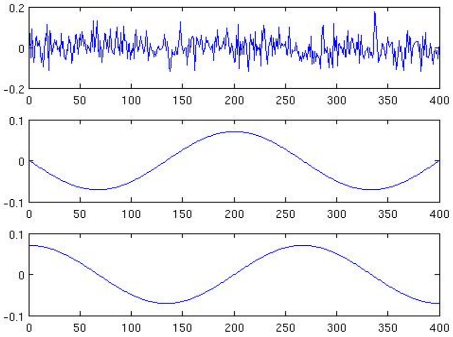

Figure 2.13: The columns of U from the GSVD. The noise has been isolated in a separate vector, and the remaining signals are noise free, representing the clean circle.

Chapter 3

Experiments

The main purpose of this study was to determine the effectiveness of Dettinger's component resampling method in terms of estimating a probability distribution function for future predictions based on a given set of predictions, as well as to explore ways that the method might be improved. The point of Dettinger's method is to create a large number of data sets which capture the original data and can be used to make a histogram which estimates a smooth probability density function for the data.

3.1 Toy Data Sets

Our initial goal was to test how well Dettinger's method was able to reconstruct a probability distribution based on a small sample of data. To do this, we created a number of artificial data sets, which we will refer to as "toy" data sets. For each time step Matlab was used to create a histogram to describe the probability distribution. We then randomly selected n points from each histogram to create the corresponding time step in each of the n artificial time series. After running Dettinger's method on these toy data, we looked to see if the histograms this created were similar to the original histograms. We also compared the histograms from Dettinger's method to histograms using only the original n randomly selected data points at each time step. If Dettinger's method works well, the histograms created using thousands of reconstructions should match the original, user-created histograms better than the histograms which use only n time series.

3.1.1 Reduced SVD

A rule of thumb for the SVD is that you can capture 90 percent of the variance in a data matrix X by retaining enough dimensions so that the normalized singular values sum up to 0.90. Remember that the singular values are the diagonal entries in the Σ matrix from the SVD, and are found by taking the square root of each of the eigenvalues of XX^T. By dimensions we mean the number, d, of columns in matrices U and V, and entries in Σ, which are retained when reconstructing the data $X_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}$ with only the first columns of the individual matrices via $X_{m\times n} = \hat{U}_{m\times d}\hat{\Sigma}_{d\times d}\hat{V}^T_{n\times d}$. To find the d necessary to to keep 90 percent of the variance in the original data you simply have to normalize the singular values (divide each one by the sum of all the singular values) and then add them up one by one, from largest to smallest, until the sum is at least 0.90. The number of singular values necessary to reach this sum is d.

Figures 3.2 through 3.5 are examples from toy data sets which had ten time series of seven time steps each. Since the data matrix for each data set is 7×10, its rank is at most seven, and therefore it can be fully described by a basis in $\mathbb{R}^7$. In other words, a basis $B_{7\times 7}$ with a coordinate matrix $K_{7\times 10}$ can fully represent the data matrix $X = BK$. Each row in the figures gives information on a different time step. Histograms from the original probability distribution which the data comes from are shown in the second columns as a measure of how well Dettinger's method is performing. In the third column, a histogram of the ten data points is given for each time step. The first column on the left shows histograms of 5,000 reconstructed data points found using Dettinger's method, except using the SVD to create the component vectors. The column farthest to the right contains histograms of 5,000 reconstructed data points created using Dettinger's method exactly as described in his paper [2]. Histograms of just the original ten data points are shown in the third column from the left.

Notice that the histograms in the first column of figure 3.2, which were created using only five of the seven possible dimensions, are very similar to those in the fourth column, which were created using Dettinger's original method, with all seven dimensions. In fact, even using only one dimension gave a good estimate, as shown by figure 3.3. This is not always the case, however, as seen in figure 3.4 which shows a 1-dimensional reconstruction of a different data set. This second data set does not do a very good job of representing the reconstructions using all dimensions (shown in column 4).

The reason we are looking at this is that sometimes it is useful to remove the last few dimensions in an SVD reconstruction since this can remove unwanted noise.

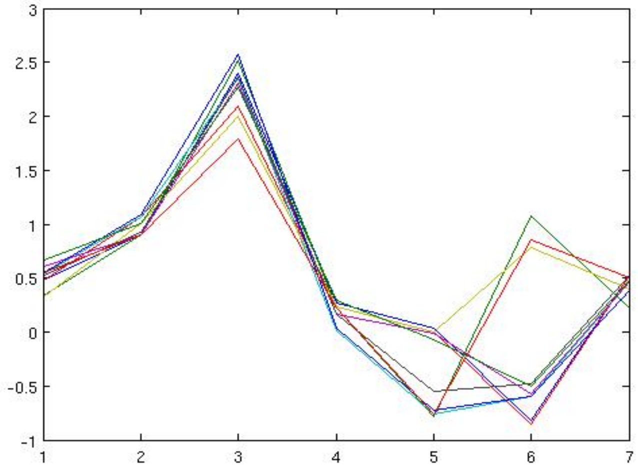

Figure 3.1: Toy data set of 10 forecasts spanning 7 time periods. This is similar to the data used for the histograms in figures 3.2 through 3.5

Bimodal distributions were not usually captured well using Dettinger's method.

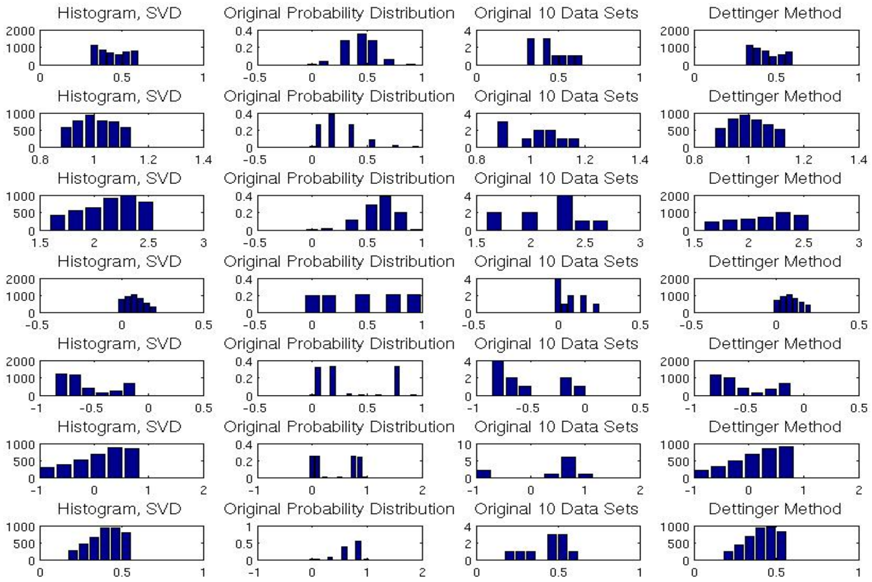

Figure 3.2: The left-hand column shows data from reconstructions which used the SVD of the 10 forecasts over 7 time steps (one time step in each row). In this case, 5 dimensions were used, which will retain at least 90 percent of the variance in the data. Removing some of the information did not change the results much from the right-hand column's graphs.

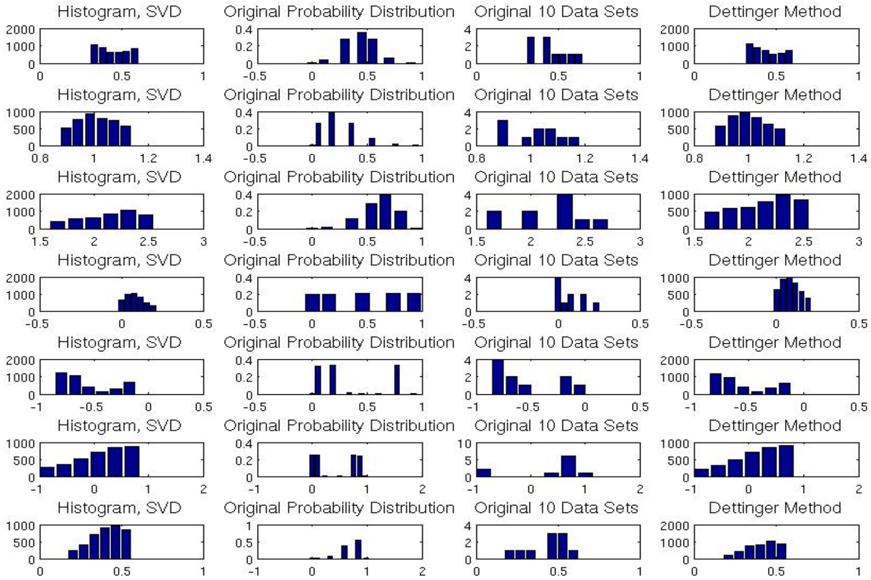

Figure 3.3: The reconstructions represented by the histograms in this figure were created using only 1 dimension from the SVD. The probability distribution is very similar to that in figure 3.2, despite the fact that much of the data's variance was removed.

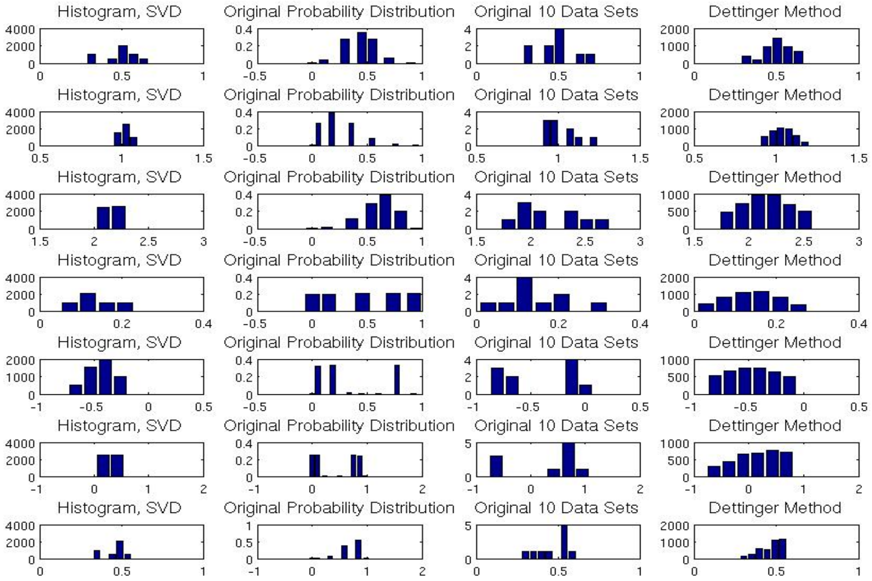

Figure 3.4: The data shown here is from a different data set than figures 3.2 and 3.3, but still containing 10 forecasts with 7 time steps. Only 1 of 7 dimensions of the SVD of the original data was retained. Unlike in figure 3.3, this resulted in very different probability distributions than those in the last column, which were created using all of the component vectors in Dettinger's original method. It is therefore possible for probability distribution functions created with information from a reduced SVD to vary significantly from those created using all component vectors. For this case, Dettinger's original method appears to match the original distribution functions more closely than the reconstructions from only 1 dimension.

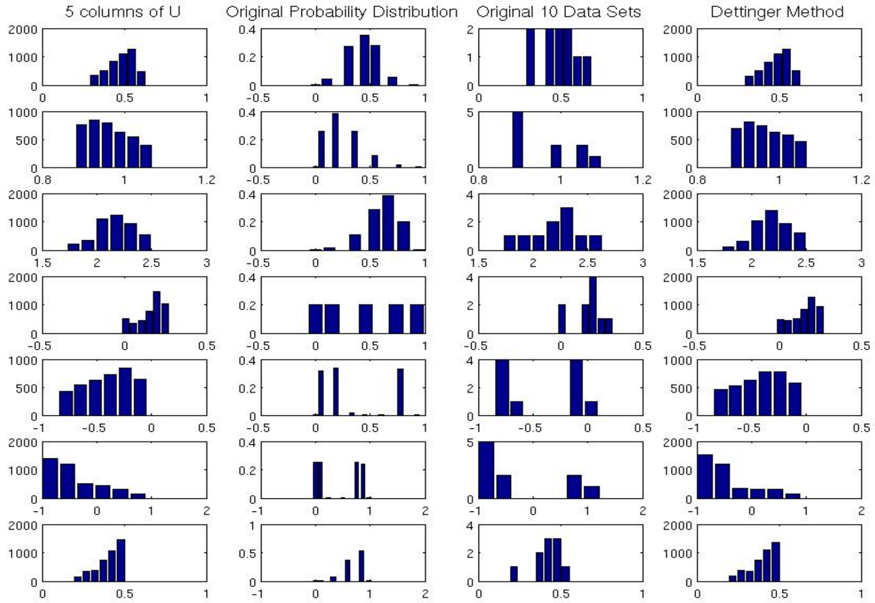

Figure 3.5: These histograms show another data set from same original probability distribution as figure 3.2 (Ten time series of 7 time steps each). The reconstructed data in the left-most column comes from using the SVD with only 5 columns, capturing 90 percent of the variance. The resulting distributions are similar to those in the right-hand column, which used Dettinger's original method, but there are some noticeable differences. Neither the first column of reconstructions nor the last column seem to be reconstructing the original distributions' shapes (second column) significantly better than the other. For this data set at least, removing noise with the SVD does not significantly change the distribution function.

In Dettinger's paper, he notes that "Because the method is based on PCA, the components of the resampled ensemble are based on its first and second statistical moments, so that the resulting smoothed PDFs [probability distribution functions] tend toward Gaussian shapes, However, that tendency is relatively weak" [2]. Bimodal distributions are definitely possible to represent, as can be seen by comparing the histograms in the fifth row of figure 3.2. The histogram found using Dettinger's method creates a bimodal distribution similar to that of the original probability distribution. However, the trend toward Gaussian shapes can also significantly distort the original distributions, as in the second to last row of figure

3.2.

Another aspect of these graphs which is important to note is the difference between the third column from the left, which shows the results of creating a histogram using only the ten original data points, and the two outside columns which contain information from 5,000 reconstructed data points. Dettinger argues that the histograms created from the large number of reconstructions provide more detailed information about the original probability distribution than the histograms with only the original data. For some cases, such as the second, fifth, and seventh rows of figure 3.2, this appears to be true. For some of the other time steps, however, it is difficult to assess which histogram is the most accurate. What is definitely true is that the histograms from the large numbers of reconstructions are less choppy than those which use only ten data points. Because of this, we could approximate a continuous probability density by interpolating a smooth curve through the high points of these histogram bars, and this curve would match the shape of the histogram very closely rather than cutting out a lot of information.

Experimenting with skewness

Figure 3.6 shows a toy data set in which all the probability histograms are skewed to the right (with a long tail on the right) along with the results from Dettinger's method. Oddly, although three of the distributions seem to be modelled reasonably well, the second and third distributions in figure 3.6 are skewed the opposite direction. It's difficult to say whether this is due to a problem inherent to this method for estimating distributions, or whether it is because of how we are creating the toy data sets. The distributions of the original ten data sets (in the third column) do seem to match the reconstructed distributions closer than the original distributions do, suggesting that the poor fit is at least partially due to a sample which happens to not be very representative of the original distribution.

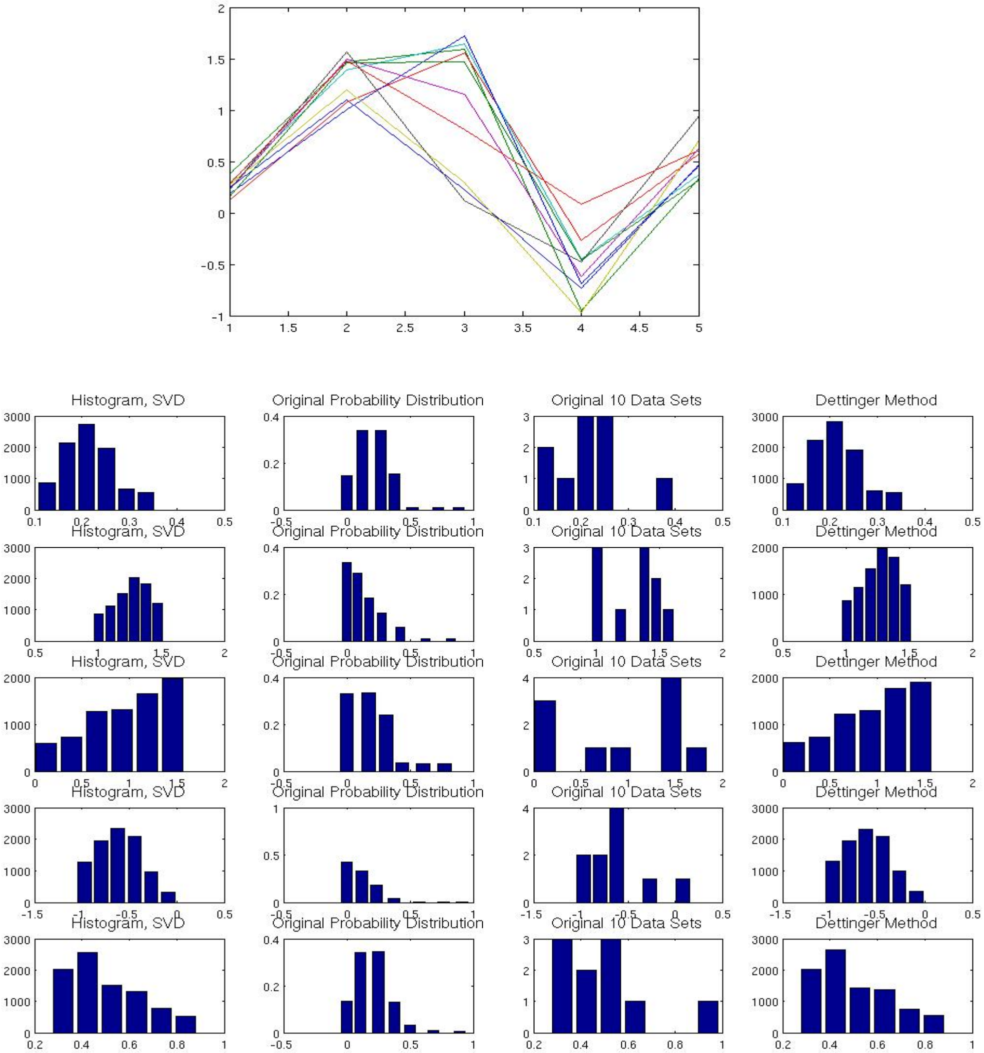

Figure 3.6: This toy data set contains ten forecasts with five time steps. The distributions at each time step (x value) were all set up to skew to the right. The original forecasts are shown above the histograms. The first and fifth time steps (top and bottom rows) followed the original skewness fairly well. The third time step (shown in the middle row) is skewed in the opposite direction as the original probability distribution, but this appears to be due to the fact that the ten data points in the sample happen to be skewed left as well.

Finding the Original Mean, Standard Deviation, and More

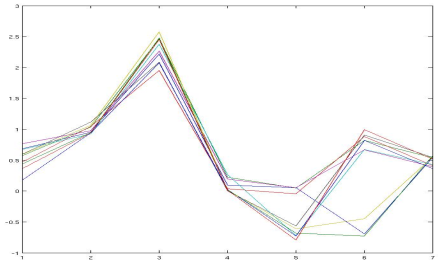

Figure 3.7: This toy data set is similar to those used in section 3.1.1 to test how well our variations on Detttinger's method create data samples with statistical moments that are similar to those of the original probability distributions.

In this section we attempt to gain a more qualitative idea of how well the reconstructions from Dettinger's method are able to reproduce the original probability distribution of a data set. Recall that for the toy data we created sets of probability distributions in Matlab, one for each time step in our artificial time series. In order to discern how well the moments of these original distributions are recovered via Dettinger's method, we first calculated the mean, standard deviation, skewness, and kurtosis of the original probability distributions we created in Matlab. Next, we performed multiple runs in which a set of data was randomly selected out of the probability distributions (Each time step in a toy data vector is a random sample from the corresponding distribution, and ten data vectors were created for each run). In each run, the sample mean, standard deviation, skewness, and kurtosis of the set of data were calculated for each time step. We also calculated statistics for 5,000 reconstructions of the data using Dettinger's method. The reconstructions were either created using a reduced SVD as described in section 2.1.5 or using the original method as described in 1.2.1. For example, table 3.1 shows the mean calculated at each time step for one set of ten time series. These statistics' relative difference from the original mean and standard deviations of the probability distributions for the corresponding time steps were calculated using Error= (New-Original)/Original. For each run, the average difference across all time steps was calculated to give an overall view of how well Dettinger's method was able to recreate the original statistics.

Means (or Sample Mean) For Each Time Step				
Time	**Original Distributions**	**Ten Data Points**	**Reconstructions, RSVD**	**Reconstructions, Dettinger**
1	0.4903	0.5401	0.5430	0.5116
2	4.0771	4.2745	4.2710	4.3024
3	9.1573	9.1514	9.1456	9.1537
4	10.5277	10.4240	10.4433	10.2477
5	10.7747	10.7893	10.8027	10.7300
6	12.1841	12.8907	12.9173	12.9132
7	9.7774	9.9200	9.9209	9.9334

Table 3.1: The sample mean taken on the toy data in figure 3.7. Note that all of the methods shown tend to be biased in the same direction, and are giving comparable results.

Table 3.2 compares the means and standard deviations of the original probability distributions to those of a sample of data, as well as those of 5,000 reconstructions from using Dettinger's method on this sample data. The data consists of ten time series, each seven time steps long. The sample data is similar to that in figure 3.7, but is slightly different for each run. The original distributions are the same for each run, with means for the original probability distributions running from 0.5 to 12.2 and standard deviations between 0.13 and 1.93. The first row of numbers compares the means of the original probability distributions to the sample means of the ten data points randomly selected from those distributions. The number shown under each run is the average of the relative difference between these two means across all seven time steps. The next two rows do the same type of comparison, but now the sample mean is that of the 5,000 reconstructions from Dettinger's method, either created using a reduced SVD as described in section 2.1.5 or using the original method as described in 1.2.1. The reduced SVD uses two or three of the seven columns of U, whichever is sufficient to capture approximately fifty percent of the information. The last three rows are the same as the first three except comparisons were made between the standard deviations rather than the means.

Relative Difference Between Means and Standard Deviations			
Data Sets, Statistic	**Run 1**	**Run 2**	**Run 3**
Original and Sample, mean	0.0305	0.0181	-0.0185
Original and RSVD, mean	0.0319	0.0182	-0.0183
Original and Dettinger, mean	0.0205	0.0215	-0.0180
Original and Sample, s.d.	0.0673	-0.0739	-0.0647
Original and RSVD, s.d.	-0.1168	-0.2338	-0.2176
Original and Dettinger, s.d.	-0.0009	-0.1216	-0.1220

Table 3.2: The relative error between the means (top three rows) or standard deviations (bottom three rows) of either the sample data and the original probability distribution or 5,000 data points from resampling and the original probability distribution. The mean of the error is taken across seven time steps, as described above. Original = artificial distributions. Sample = Sample of 10 data points at each time step taken from the original distribution. RSVD = Reconstructions from a version of Dettinger's method with a reduced SVD to define the component vectors. Dettinger = Reconstructions using Dettinger's method as presented originally. We did not observe any patterns in the bias. Both the RSVD and Dettinger's original method gave similar results for the mean, but the standard deviation error was larger using the RSVD.

Relative Difference Between Skewness and Kurtosis			
Data Sets, Statistic	**Run 1**	**Run 2**	**Run 3**
Original and Sample, skew	-6.0096	-4.5726	-2.2455
Original and RSVD, skew.	-1.5707	-6.5769	-2.4673
Original and Dettinger, skew.	-2.2154	-3.9354	-2.9006
Original and Sample, kurt.	-0.0288	-0.1424	0.1752
Original and RSVD, kurt.	0.1923	0.3858	0.1028
Original and Dettinger, kurt.	0.2743	0.3765	0.2161

Table 3.3: Same as table 3.2, but comparing skewness (asymmetry) and kurtosis (peakedness). Skewness was always underestimated (as denoted by the negative sign). The RSVD and Dettinger's original method gave comparable results.

There does not appear to be a consistent pattern in which method matches the original mean the best. The standard deviation of the sample data usually matched that of the original distribution the best, while the RSVD provided the least faithful reconstruction of the standard deviation. It's interesting to note that even using only 3 out of 7 columns of the SVD results in a reconstructed set of time series with a mean which is within three percent of the original mean. Also, the standard deviation of the sample data and its reconstructions was almost always less than the original deviation. This makes sense, since it would be difficult to capture all the deviation of the original distributions using only 10 data points.

On average, skewness was not reconstructed well. As shown in figure 3.3, the skewness of the reconstructions was as much as 600 percent off from the original skewness, and was always less than the original skewness. This matches the problems with reconstructing skewness that we saw with the histograms of some of the toy data sets. The match in terms of kurtosis was much better, staying within about 40 percent of the original.

Overall, there does not seem to be a clear pattern in whether using Dettinger's method with a reduced SVD, the original version of Dettinger's method, or even just the initial data sample creates a more accurate representation of the original probability distribution in terms of the first few moments of the distribution. This will be discussed more at the end of section 3.1.2.

3.1.2 GSVD

We ran several examples with an altered version of Dettinger's method which utilizes the GSVD. As described in section 2.2.2, the GSVD can be used to create an ICA factorization of the form $X = SA$, where the columns of S (or U, using our GSVD notation) form an orthonormal basis for the data in X. To adapt the GSVD to Dettinger's method, we used S for Dettinger's E matrix, and A (CB^T from the GSVD notation) for his P^T. The results of using this method on a toy data set similar to that in figure 3.7 are shown in figures 3.8 through 3.9. The first graph

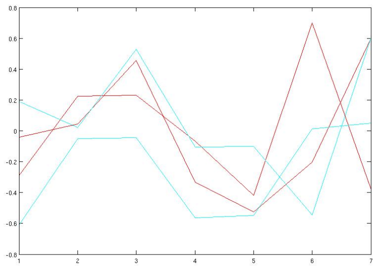

Figure 3.8: The first two columns of E using the SVD (red) and GSVD (blue) are different, despite describing parts of the same subspace which is spanned by the original data.

shows the first two columns of E from each method (remember that these are from a normalized data set, so the mean is 0 and the standard deviation is 1). For the sake of neatness and clarity, only the first two vectors from each method are shown. Showing all five vectors does not give any additional information as none of the vectors of the two bases overlap with each other. It is immediately obvious from this graph that the basis vectors created by the two methods are different, despite both forming bases for the same vector space. Recall that the basis from the SVD was constructed with the goal of capturing the greatest amount of error in the first vectors of the basis, whereas the basis from the GSVD was designed to separate out independent signals.

As was shown in the noisy circle example, the noise is located in the first columns of U, so to remove noise we remove the first column of U instead of removing columns from the end of U as with the reduced SVD. Figures 3.9 through 3.12 compare histograms using the reduced SVD and reduced GSVD in Dettinger's method. They show that the reconstructions from the GSVD are slightly different from those reconstructed using the SVD, but the GSVD does not appear to model the original distributions any better than simply using the SVD.

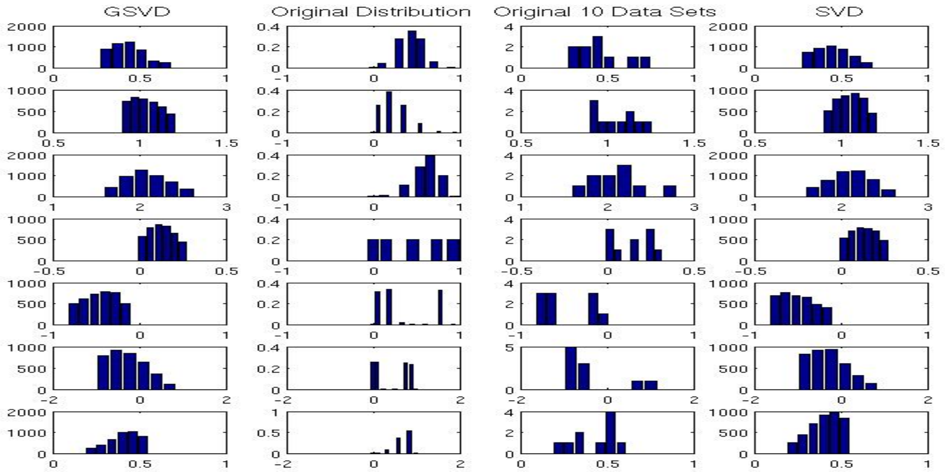

Figure 3.9: Toy data set of ten forecasts with seven time steps, using 7 of 7 columns from the components matrix. Although there are slight differences between reconstructions from the SVD and those from the GSVD, neither method seems to consistently model the original distribution function best. Also, as can be seen in the first row, direction of skewness is heavily influenced by the sample data (shown in the third column).

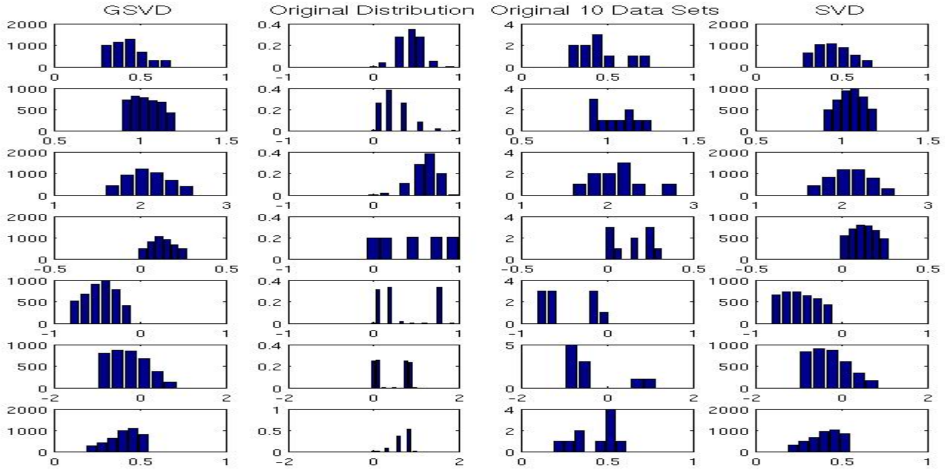

Figure 3.10: The reconstructions used to form these histograms come from the same toy data set as figure 3.9, using 6 of 7 basis vectors (columns of U), thereby removing one noisy column. Sometimes the histograms from the GSVD are quite similar to those from the SVD (i.e. the second row from the bottom), but for other time steps the histograms have completely different shapes (i.e. the top two rows). This suggests that, in some cases, different components are being removed as noise by each method.

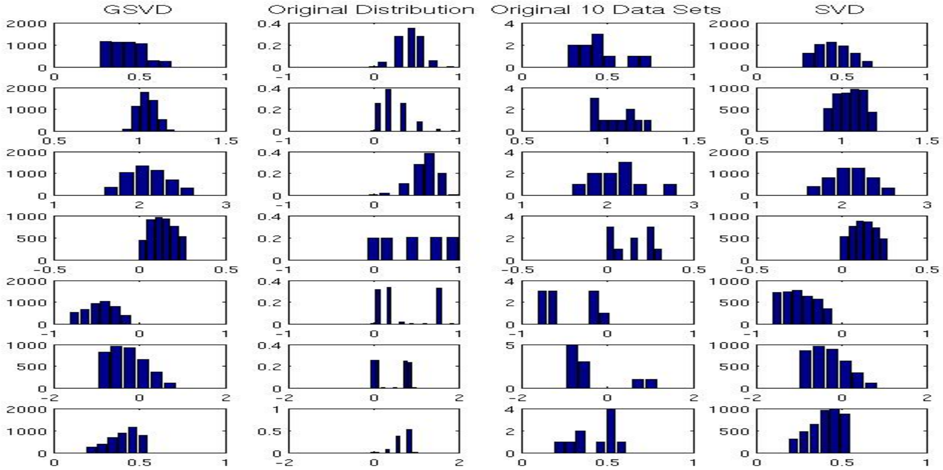

Figure 3.11: Same toy Data Set as figures 3.9 and 3.10, results using 5 of 7 columns. Compared to figure 3.10, removing more noise led to some histograms matching the original distributions' shapes more closely, while others' shapes matched less well than when only one column was removed.

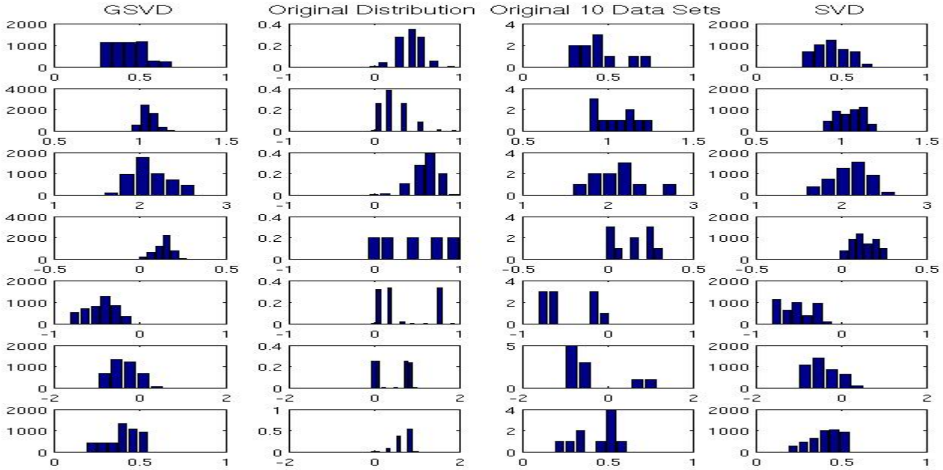

Figure 3.12: Toy data set of 10 forecasts with 7 time steps; results using 3 of 7 columns of the component matrix. At this point there are significantly fewer possible reconstructions, as discussed in section 2.1.5. For the SVD, this resulted in some distributions with multiple peaks (rows 4 and 5), while the GSVD stayed bell shaped. This suggests that the SVD may reconstruct non-unimodal distributions more accurately than the GSVD, although the distributions are still not very close to the original distributions.

More Statistics, Including the GSVD This Time

Table 3.4 compares the statistics of various data sets to the original toy probability distributions. The data being compared are: a random sample of data from the probability distributions, 5000 reconstructions using Dettinger's method with the SVD, 5000 reconstructions using Dettinger's method and only the first 3 columns of U from the SVD, 5000 reconstructions using Dettinger's method where the component vectors were found using the GSVD, and 5000 reconstructions using Dettinger's method with only the last 3 columns of the GSVD as component vectors. Dettinger's method with the full SVD is, for all intents and purposes, the same as using Dettinger's original method, as discussed earlier.

Similarly to the results in section 3.1.1, means and standard deviations were approximated to within a few percent, kurtosis was within about twenty percent, and skewness was changed significantly. Again, we found no clear pattern concerning which method best reproduced the original distribution. It does seem that the full GSVD did a much better job of reproducing the original standard deviation than its reduced version, but this was not true for other moments.

Before rescaling the forecasts (Step 6 of Dettinger's method), the mean of the reconstructions using the RSVD was generally further from zero (the mean of the normalized data set) than the reconstructions from the GSVD. Also, the standard deviation was further from one. This suggests that the RSVD does not fall into the trap of creating bell-shaped curves which exactly match the moments of the sample data as easily as the GSVD does. This is supported in figure 3.12, in which only histograms from the RSVD have multiple peaks.

The skewness measures for run 3 are interesting, in that all versions of Dettinger's method improved the estimation of skewness (The high relative errors are partially due to the fact that skewness values are very small). Also, removing noise from the GSVD for this run improved the approximation greatly.

Overall, it appears that the accuracy of one method or another, in terms of reconstructing the original moments, is either highly dependent on the sample data or completely random.

Table 3.5 summarizes the results of 50 runs, each run taking a random sample of ten data points from each distribution. On average, the original sample estimated the original standard deviation and kurtosis the closest. The reduced SVD was the most successful at capturing the mean and skewness. The differences between methods are not very significant, however, because the values in each row are all within one standard deviation of each other. According to this information, none of the methods tested perform significantly better than any other when it comes to reconstructing the original distributions. This is somewhat to be expected, because since Dettinger's method uses only the sample data as input there is a limit to how much information Dettinger's method can give which is not already available in the original sample.

The fact that all of the deviations from the original moments were comparable means that the moments of each data set (sample and reconstructed) were similar to each other. This shows that, even though Dettinger's method uses artificially reconstructed data, it does not significantly alter the first few moments of the data it is reconstructing (on average). This means that Dettinger's method does not greatly distort the statistical information which it is given when it makes its large number of reconstructions. As shown by the histograms earlier, the reconstructions were not always very accurate, but what was necessarily true was that the reconstructed histograms were less choppy and could be easier interpolated with a smooth curve. Dettinger's method cannot create new information, but it can approximate a higher resolution of information, and therefore smoother estimated distribution functions, than could possibly be created using a histogram with only a few data points.

Relative Difference of Statistics			
Data Sets Being Compared	**Run 1**	**Run 2**	**Run 3**
Mean			
Sample	**0.0382**	0.0366	0.0099
Full SVD	0.0387	0.0361	0.0096
Reduced SVD	0.0383	0.0372	0.0100
Full GSVD	0.0386	0.0357	**0.0088**
Reduced GSVD	0.0395	**0.0340**	0.0108
Standard Deviation			
Sample	0.0264	**-0.0325**	**-0.0203**
Full SVD	-0.0303	-0.0801	-0.0724
Reduced SVD	-0.1680	-0.1937	-0.2280
Full GSVD	**-0.0069**	-0.1239	-0.0504
Reduced GSVD	-0.1680	-0.1937	-0.2280
Skewness			
Sample	4.7310	2.3988	-6.5806
Full SVD	**-0.2024**	-1.0789	-2.1842
Reduced SVD	3.8239	**-0.9581**	-2.7388
Full GSVD	0.6876	-2.2910	-4.5273
Reduced GSVD	1.8442	-3.0983	**-1.7351**
Kurtosis			
Sample	**0.0085**	**-0.2079**	-0.0423
Full SVD	0.2918	0.2730	0.2750
Reduced SVD	0.2490	0.2345	0.2111
Full GSVD	0.1871	0.2400	0.2624
Reduced GSVD	0.1500	0.1867	**-0.0159**

Table 3.4: We compared the estimates of the moments from different samples in order to determine whether any of the versions of Dettinger's method we've discussed are better overall. See section 3.1.1 for details on how these numbers were calculated. The sample data is similar to that in figure 3.7, but is slightly different for each run. Each number is the mean relative difference between the moments of a data set and the original distribution. The mean of the difference is taken across 7 time steps. The original distributions are the same for each run. The smallest difference for each statistic for each run is in bold. The fact that no method consistently provided the smallest error suggests that there is no significant difference between using one method or another, and that if one method does match the original distribution more closely than another it is either by random chance or due to some unnoticed, subtle difference in the sample.

3.1.3 Probabilities in 3D

The main goal behind using Dettinger's method is to get estimates of probability distributions from even a small sample of data points. We have already

Moment	**Data Set**				
	Sample	**SVD**	**Reduced SVD**	**GSVD**	**Reduced GSVD**
Mean	0.0118	0.0119	**0.0116**	0.0125	0.0120
Std. Dev.	**-0.0149**	-0.0651	-0.1903	-0.0699	-0.1903
Skewness	-1.5135	-0.6307	**-0.6094**	-0.7534	-0.7027
Kurtosis	**0.1507**	0.2828	0.2198	0.3070	0.2549

Table 3.5: Summary of Statistics for 50 runs. Numbers are the average difference between the labelled data set's and the original probability distributions' moments (mean, standard deviation, skewness, and kurtosis), given by (New-Original)/Original. The smallest value for each statistic is in bold. On average, mean and kurtosis were overestimated while standard deviation and skewness were underestimated. The reduced SVD and reduced GSVD underestimated the standard deviation by the greatest amount, suggesting that we may have removed too many component vectors, and started cutting out variance which was inherent in the data rather than being noise. The fact that no single method had the smallest error for all moments, and that the values in each row are all within one standard deviation of each other (not shown) shows that no method consistently provides a more accurate approximation of the original distribution than the others.

done this with histograms, but that's not a very readable format for a time series where we're concerned with a different probability distribution at each time step. Also, some applications of Dettinger's method call for more information than can be given by a simple histogram. For example, suppose we want to find 75 percent confidence intervals about the sample mean for each time step. This would mean the interval of possible values containing the sample mean in which 75 percent of

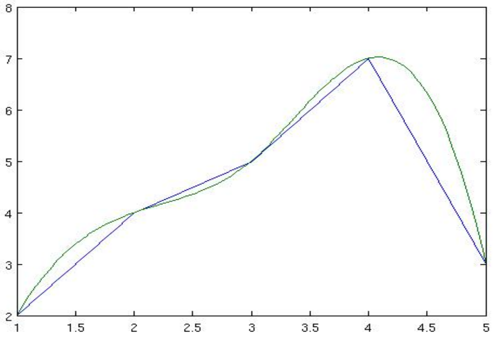

Figure 3.13: Example of cubic spline interpolation on five data points.

the probability density is located, which corresponds to the interval over which there is a 75 percent chance that the true value is located. To find such a confidence interval, we must either use the area in each bar of the histogram or estimate the continuous probability density from the histogram and then integrate. The first method restricts us to a finite number of possible intervals, however, since I can only know the area between the edge of one bin and the edge of another.

For the second method, finding a continuous function to describe the distribution, we must interpolate between the height of each bar. There are many ways to do this. For example, a best fit curve could be found for a polynomial using linear least squares. This leads to the problem of choosing what degree polynomial to use, however. Besides, this type of curve fitting makes it so that the data points don't necessarily lie on the fitted curve.

A method of curve-fitting that does include the data points is cubic splines. Figure 3.13 shows an example of five data points with both a linear spline (straight line) and cubic spline interpolation. A cubic spline $S(x)$ of n data points (x_1, y_1) to (x_n, y_n) is a set of cubic polynomials, $S_1(x)$ to $S_{n-1}(x)$, each being defined only over a unique interval between two adjacent data points. The splines are designed so that $S(x)$ is continuous and twice continuously differentiable between x_1 and x_n. In this way, $S(x)$ provides a smooth curve connecting all of the data points.

We use Matlab to create cubic spline interpolations of the histograms and then antidifferentiate the resulting polynomial to find the integral, and therefore the cumulative probability density. Figure 3.15 shows the normalized probability distribution (blue) and cumulative integral (red) for the toy data set shown in figure 3.14. These integrals were then used, using Matlab's meshgrid and surf functions, to create figure 3.16 which shows cumulative probability densities at each time step using a color scale in which blue is the bottom extreme at 0 and red is the maximum value of 1.

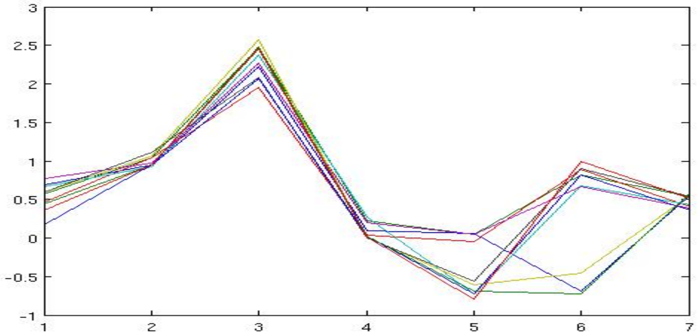

Figure 3.14: This randomly generated data was used to demonstrate how Dettinger's method can be used to estimate probability density functions, shown in figures 3.15 and 3.16.

The densities should all be between 0 and 1, but some go slightly below 0 because the spline goes out to the edges of the bins, so it is possible for the spline to be slightly below the $x-$axis at the far left of the histogram and therefore for the integral to be negative.

Figure 3.17 is similar to figure 3.16, except the colors represent the integral of the probability density between that y value and the mean. For example, the third column of colors, corresponding to time step 3, has a dark blue bar at about $y = 2.3$, meaning that the sample mean is 2.3 and there is zero probability of this mean event occurring. As you go larger than the mean, the probability of an occurrence increases more slowly than when you get smaller than the mean. Therefore, there is a higher probability density below the mean than above at this time step. This could also be seen by looking at the third graph down in figure 3.15, where you can see that the histogram is taller immediately to the left of 2.3 than it is to the right, so the probability of being just below the mean is higher than the probability of its being just above. The probability of zero at the mean is theoretically expected, since the integral from the mean to the mean must be zero. Some of the other columns don't hit zero because none of the values at which the integral was computed exactly equalled the sample mean.

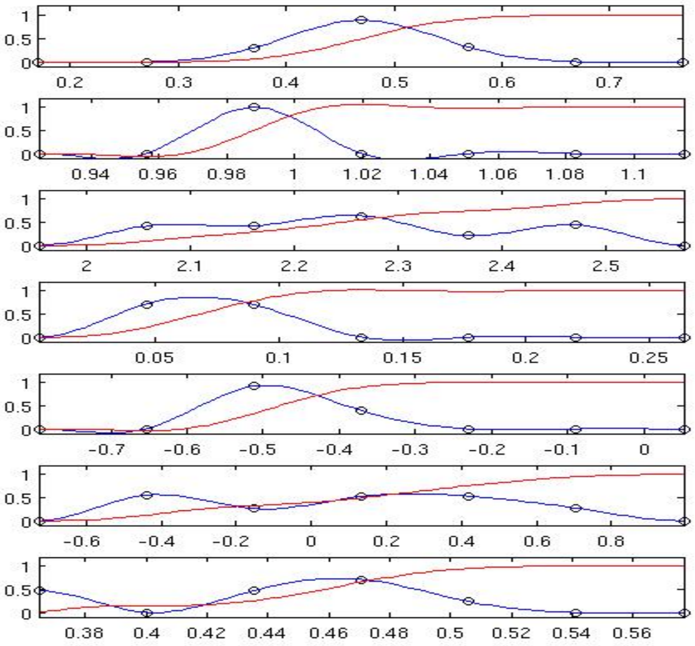

Figure 3.15: Smoothed, normalized histograms for data in figure 3.14; cumulative integral shown in red. Each graph is a different time step.

Another technique we used to represent the data distribution in three dimensions was a radial basis network interpolation of the histograms, which yielded graphs such as that shown in figure 3.18 along with its original data. Radial basis networks create a 3D surface consisting of points connected by edges, and the newrb command in Matlab builds a surface of this type which matches with the original points to within a specified least squares error goal. For figure 3.18, the colored surface is an interpolation of data in histograms like we've seen earlier. The newrb function was given time steps as x coordinates, the middle value for histogram bins as the y values, and the height of each histogram bin as z values. The $xy-$plane is shown, with colors corresponding to z values.

This could be useful for interpolating across gaps in data. Of course, if the missing data was at a maximum or minimum this would not be very accurate, but

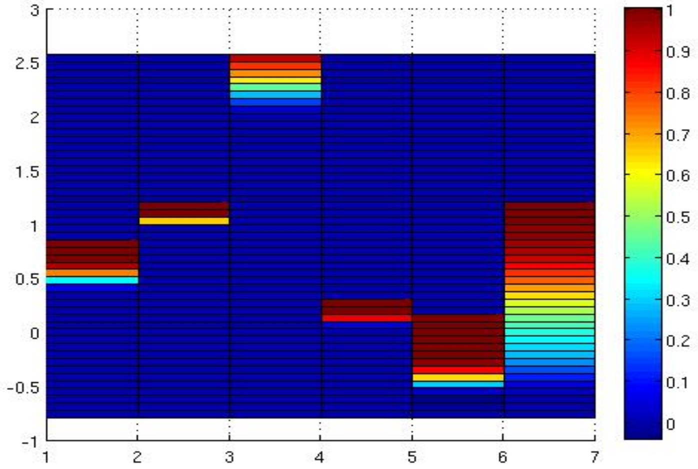

Figure 3.16: Cumulative probability distributions for figure 3.14, estimated using Dettinger's method.

for many points it would provide a reasonable estimate. Even with figure 3.18 we can interpolate in between the given data points.

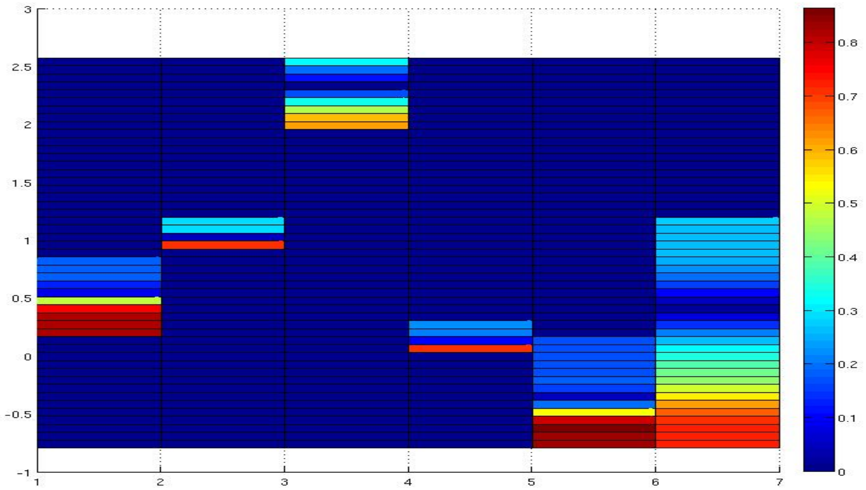

Figure 3.17: The color of each bar represent the probability of an event occuring between that bar and the mean of the data in figure 3.14. For example, the third column, corresponding to time step 3, has a dark blue bar at about $y = 2.3$, meaning that the sample mean is 2.3 and there is zero probability of this mean event occurring. As you go larger than the mean the probability of an occurrence increases more slowly than when you get smaller than the mean. Therefore, there is a higher probability density below the mean than above at this time step.

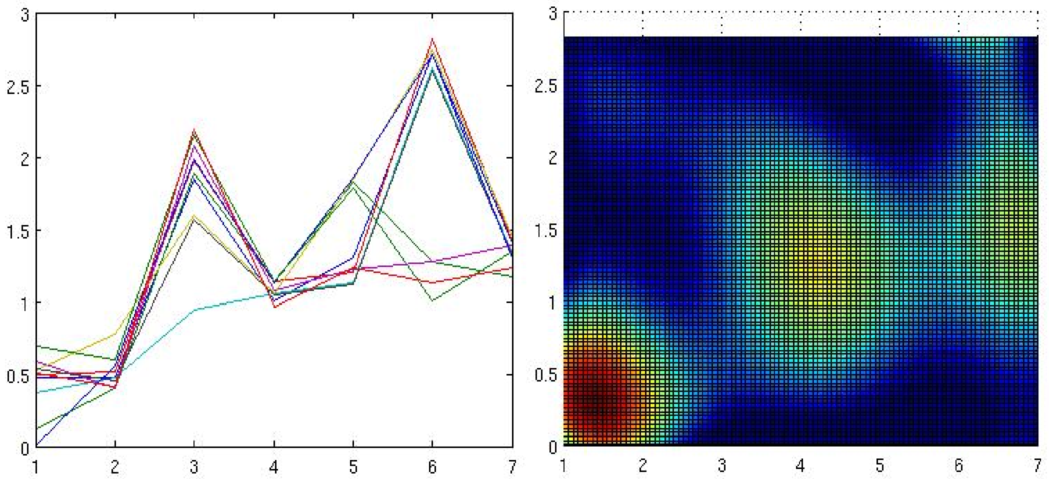

Figure 3.18: The graph on the right was created using the data shown on the left. A histogram was created using Dettinger's method, and then relative probabilities were interpolated between data points. Red means a higher probability density, while blue is a 0 probability.

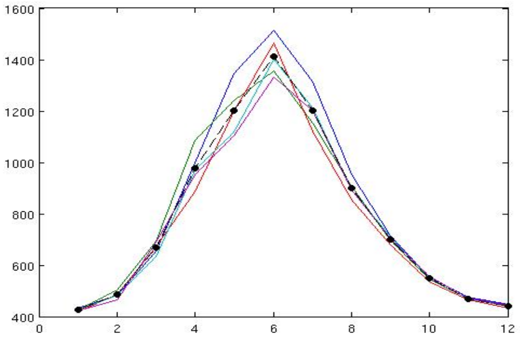

Figure 3.19: Five models for median monthly streamflow in the Consumnes River. Black circles show the mean from the original five models. Data from [1].

3.2 Applications to Actual Data

3.2.1 Streamflow Model Data

Estimating probability distribution functions using Dettinger's method and the SVD.

A set of data from streamflow models was used to experiment with the accuracy of using Dettinger's method on actual data. The models use factors such as temperature and rainfall as inputs. This provides a good example of the type of problem that Dettinger's method can be applied to, since it involves multiple time series which give slightly different predictions for the same event.

Each line in figure 3.19 represents the output from a different model used for estimating/predicting streamflow, as shown in figure 3.26. Each point on the graph shows the median streamflow for each month in cubic feet per second (cfs). The months are ordered according to water years, which run October through September instead of January through December, so "1" on the $x-$axis of this graph represents October. The medians were calculated over a 150-year period.

Figure 3.20 shows 10,000 reconstructions of the data using Dettinger's method, along with the sample mean taken over all five models at each month. In months

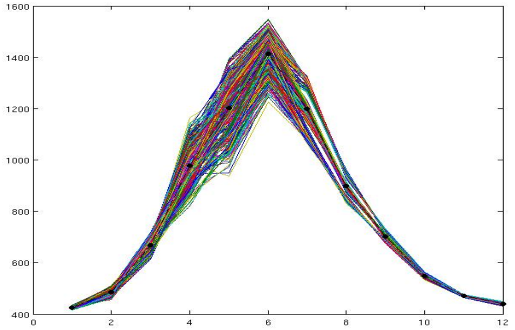

Figure 3.20: 10,000 Reconstructed models of median streamflow. Months range from 1=Oct to 12=Sept. Black circles show the mean from the original five models.

seven and eight (April and May) the separations of the models into three distinct groups is apparent, as all the lines separate into three clumps with white space in between. Looking at figure 3.21, this separation is not shown very well for month seven (the top row), but can be seen fairly clearly in month eight, which has local maxima at the far left, around the center, and in the second bar from the right. There are several possible explanations for why the model values look more separated for month seven than month eight but are not presented that way by the histograms produced using Dettinger's method. First of all, the placement of the histogram bins (the range of x values covered by each bar) might simply be a poor choice for capturing the variation in this particular data set. Also, we can't estimate the density of the lines in each clump very well simply by looking at figure 3.20, and the groupings may be less significant than they seem if some actually only contain a few lines out of the 5,000 reconstructions.

The fact that so many non-unimodal distributions are showing up (multiple peaks in the histogram instead of just one) is very promising as well as puzzling. It is promising in that it shows that Dettinger's method is capable of producing non-Gaussian distribution functions, but puzzling in that we did not see many of these non-Gaussian functions when we were trying specifically to create them using

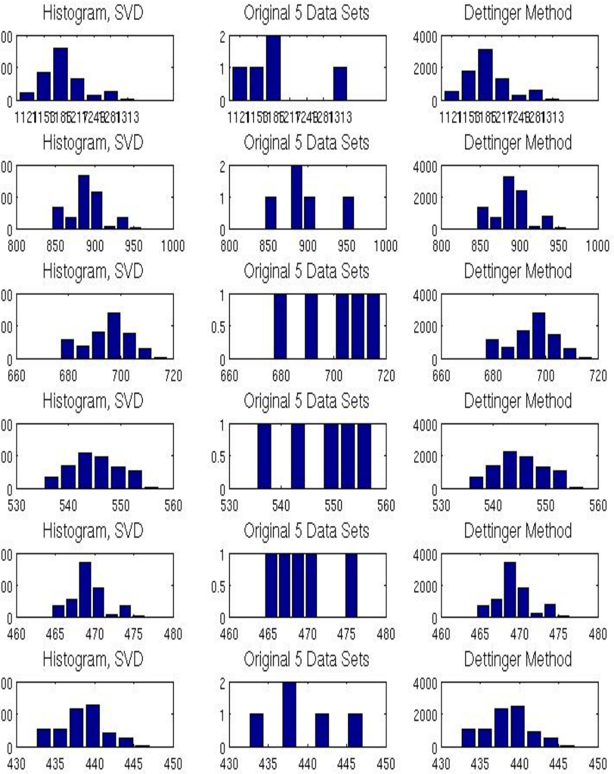

Figure 3.21: Results from using Dettinger's method on April (first row) through September (bottom row) in the streamflow data.

toy data sets. There may be a problem with how we were creating the toy data sets, or maybe there is some particularity in the data that we haven't been taking into account.

Using the SVD and GSVD to remove noise.

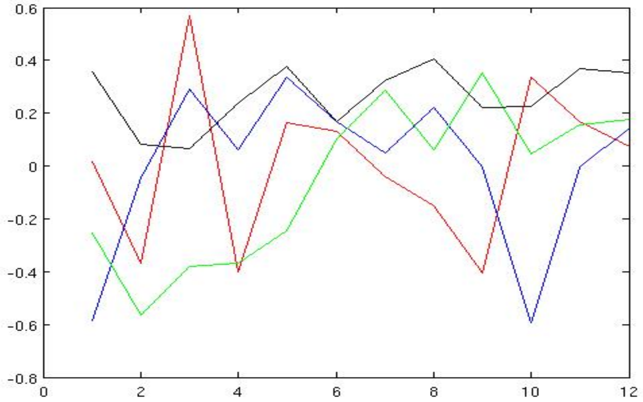

Figure 3.22: Columns of E found using the GSVD. First: red, Second: Blue, Fourth: Green, Fifth: Black. The first few columns contain the noise, and their curves are not as smooth as the others'.

Recall that in the noisy circle example the noise was in the first column of U. Figure 3.22 shows some of the columns of E from using the GSVD on streamflow data. As expected, the first couple of columns seem noisier, in that they jump around more.

Figures 3.23 through 3.25 are from the same stream data with 5 models and 12 time points (months). Histograms from using Dettinger's method with the GSVD are compared to histograms from the original data and from Dettinger's method with a reduced SVD. Often the results from using the GSVD and SVD are similar, but there are definitely greater differences between the two than we saw with the toy data sets.

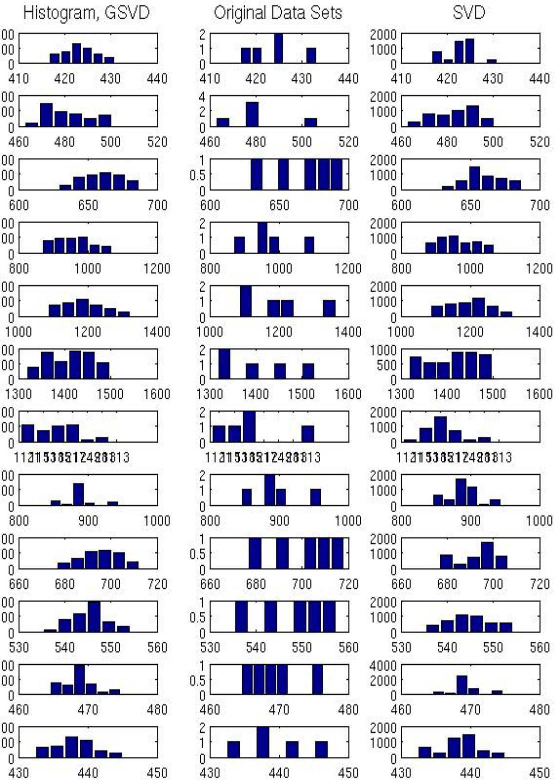

Figure 3.23: Results using 3 of the 5 columns of U. For many rows the results from the reduced SVD and GSVD are fairly different, since different information has been removed in each case. There is no known original distribution, as with the toy data, but in general the histograms in the outside columns should look like smoothed out versions of those in the middle. Most of the RSVD histograms do this more successfully than their reduced GSVD counterparts, but not all.

Comparing probability distributions to observed data.

For this particular dataset, we have the actual observed streamflow as well as the predicted streamflow. Figure 3.26 shows six different models for monthly stream discharge in California's Consumnes River in cubic feet per second (cfs), along with the observed values. The average flow for each month for the years 1951 to 2005 are shown. Figure 3.26 shows the predicted flows for 2006 to 2060, using modelled future temperature and rainfall as inputs to the model. Dettinger's method can be

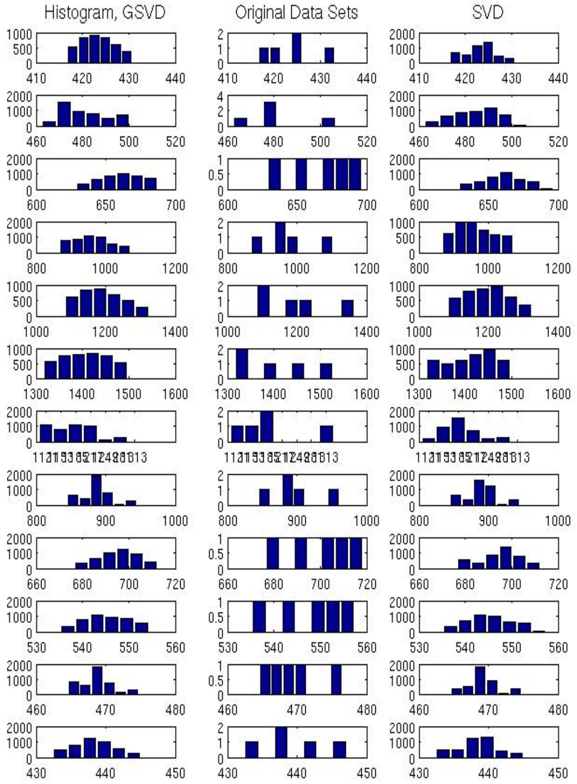

Figure 3.24: Same as figure 3.23, but removing only the first (or last) column of U from the GSVD (or SVD). The results from the GSVD trend more toward Gaussian distributions than those from the SVD. In row 2, the SVD seems to be skewed in the wrong direction.

applied to the predicted future data shown in figure 3.26 to demonstrate the probabilities of certain streamflows occurring. In order to test how reliable these probability measures will be, we can first apply Dettinger's method to the models of historical (1951-2005) data and compare the results to the observed data.

Figure 3.27 shows the relative probability density of the data for 1951-2005 from figure 3.26. It was created using cubic splines to interpolate between the histogram bars created using Dettinger's method. Figure 3.28 shows the probability of being within a certain distance of the mean, based on the normalized area under the probability density curve represented in figure 3.27. This can be viewed as

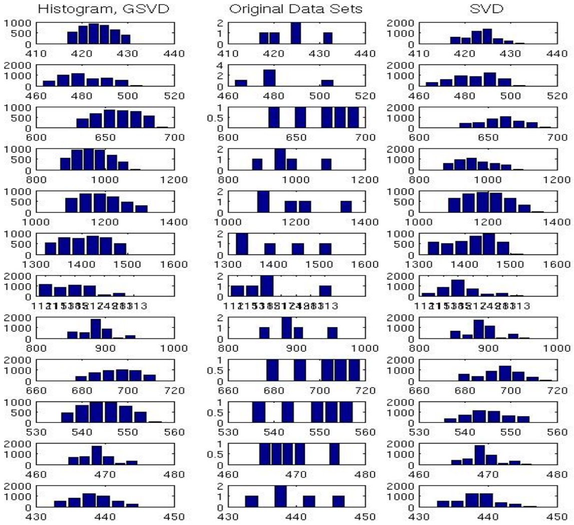

Figure 3.25: Same Data Set, Results using 5 of 5 columns.

confidence intervals. For example, there is a sixty percent chance that an event will occur between any yellow bar on the graph and the mean. The stars in both graphs show the observed data. The observed data fell within the fifty percent probability range for October (Month 1) through December and April (Month 7) through September, but the models did not predict the observed flow very closely for the remaining months. This would be important to keep in mind when deciding how much to rely on the predicted flow in figure 3.26.

3.2.2 Historical Precipitation Data

To look at the results of using Dettinger's method on larger datasets, as well as its applications with historical data rather than models, we took precipitation information for a certain weather station in the month of January over the years 1981 to 2008 [8].

Figure 3.29 shows the original data. The $y-$axis measures cumulative precipitation for the year since October, and the $x-$axis is days in January. Each

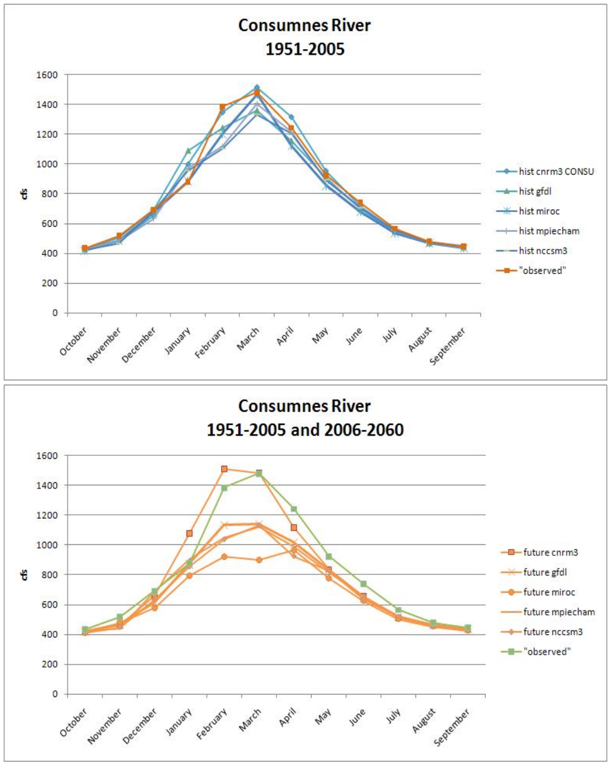

Figure 3.26: These graphs show five different models' predictions for median streamflow for each month in the Consumnes River. The top graph covers the years 1951-2005, while the bottom graph spans 2006-2060. Observed data for 1951-2005 is shown in a separate color from the modelled data in both graphs. Comparing this observed data to the predicted values for that same time period gives us an idea of how reliable the models are.

line shows data from a different year.

Dettinger's method was used to produce the histograms in figure 3.30, which were then interpolated between for 3.31 (Histograms for only ten of the 31 days are shown). Notice that the histograms all tend toward a unimodal distribution, though skewness seems to be preserved fairly well. A graph like that shown in figure 3.31 could be used to determine whether a certain day's precipitation fell within the normal range. For example, a cumulative precipitation of ten inches (measurements starting October 1st) on January fifth would be very likely, while there is only about a ten percent chance of the number being as high as 35 inches. This does not follow any commonly used statistical methods of determining confidence intervals, but it

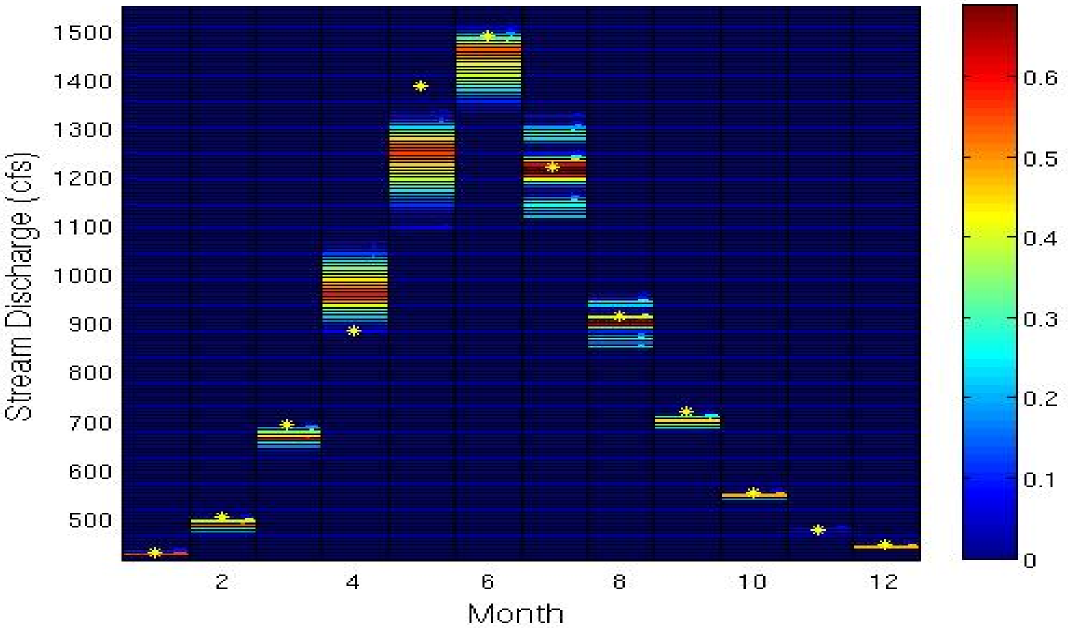

Figure 3.27: Relative probability densities based on the forecasts in figure 3.26. Red is high (tall histogram bars) and dark blue is zero or outside of the range of the reconstructions. Yellow stars are observed values.

provides a useful visualization of the historic precipitation data. The white line shows the most recent year's data, and it is easy to see that this year's precipitation fell well within the range of the most common historical precipitation levels.

3.2.3 Trends Over Time

It is important to keep in mind the limitations of this method in terms of predicting future events. For example, it does not represent means which vary over time. Suppose we wanted to use the information found above to predict a future streamflow. The monthly medians we used were for the years 1950-2100. We could say that, since the median flow for October was 424.9cfs with a small margin of error, then the median for October 2101 would likely be close to 424.9 cfs. Graphing all of those October flows, however, shows that the median flow is slowly decreasing over time, so the median flow for October of any year after 2100 is more likely to be

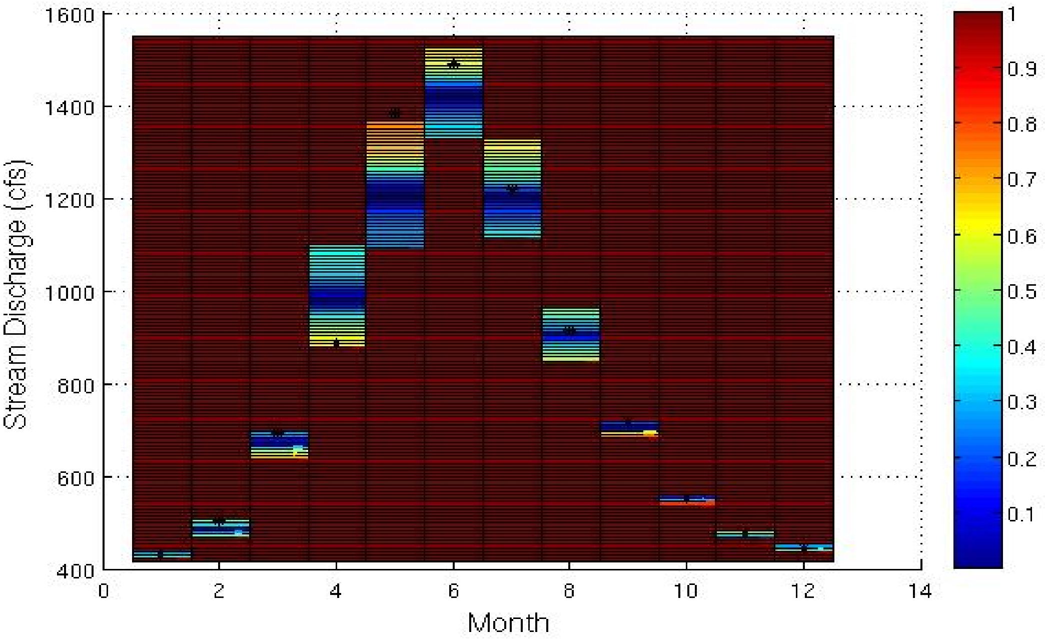

Figure 3.28: Probability of being between the bar and the sample mean of the forecasts in figure 3.26. Black stars represent observed values.

below the mean of 424.9 cfs than above it. Dettinger's method does not give us any information about that. It deals with changes over models rather than changes over time.

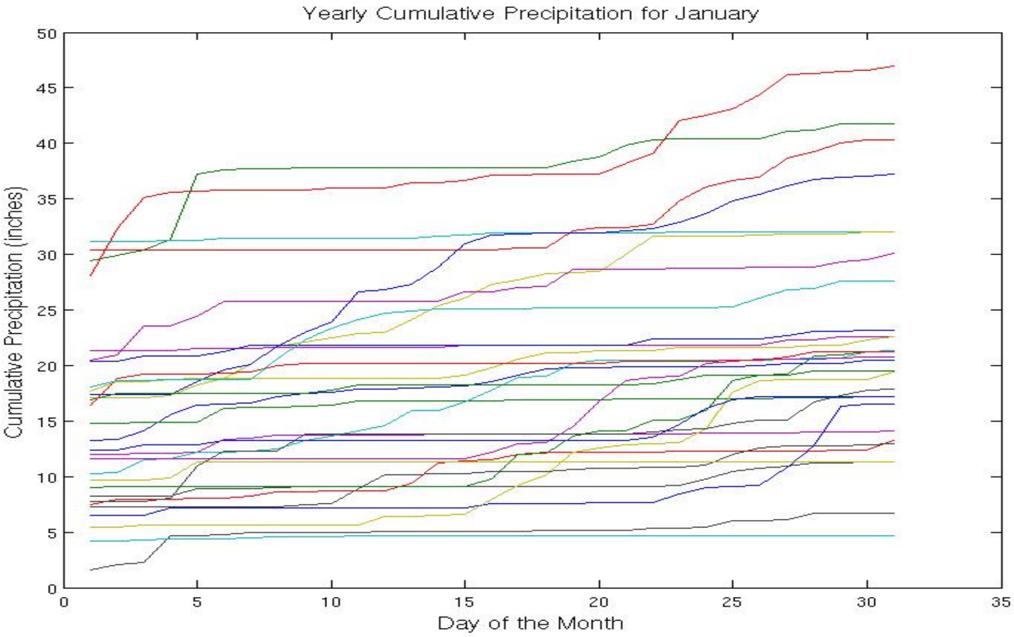

Figure 3.29: Cumulative precipitation for the month of January at a weather station in California. Each line shows a separate year.

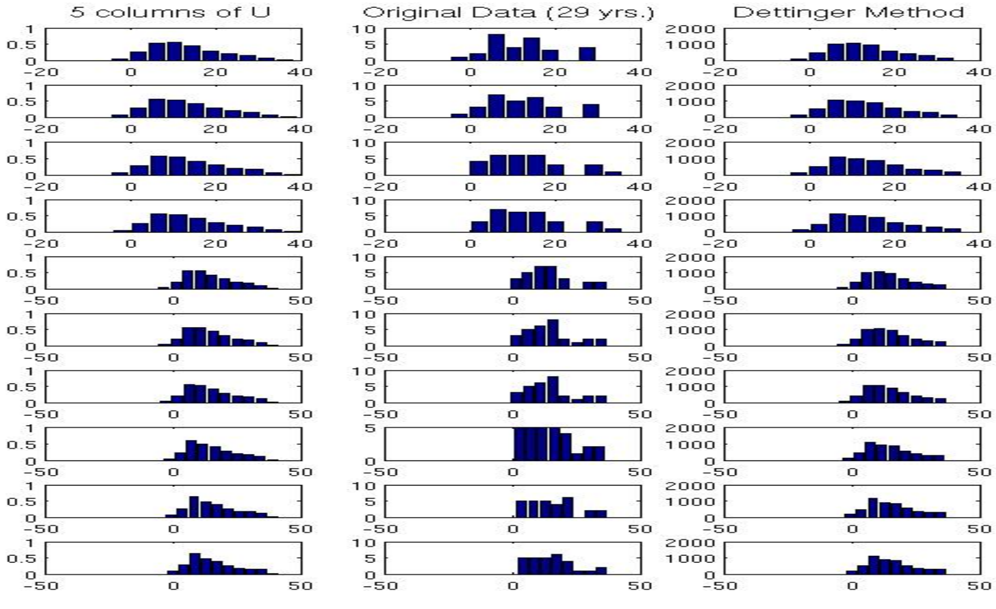

Figure 3.30: Probabilities for cumulative precipitation for January 1st (top row) through January 10th (bottom row) at a weather station in California.

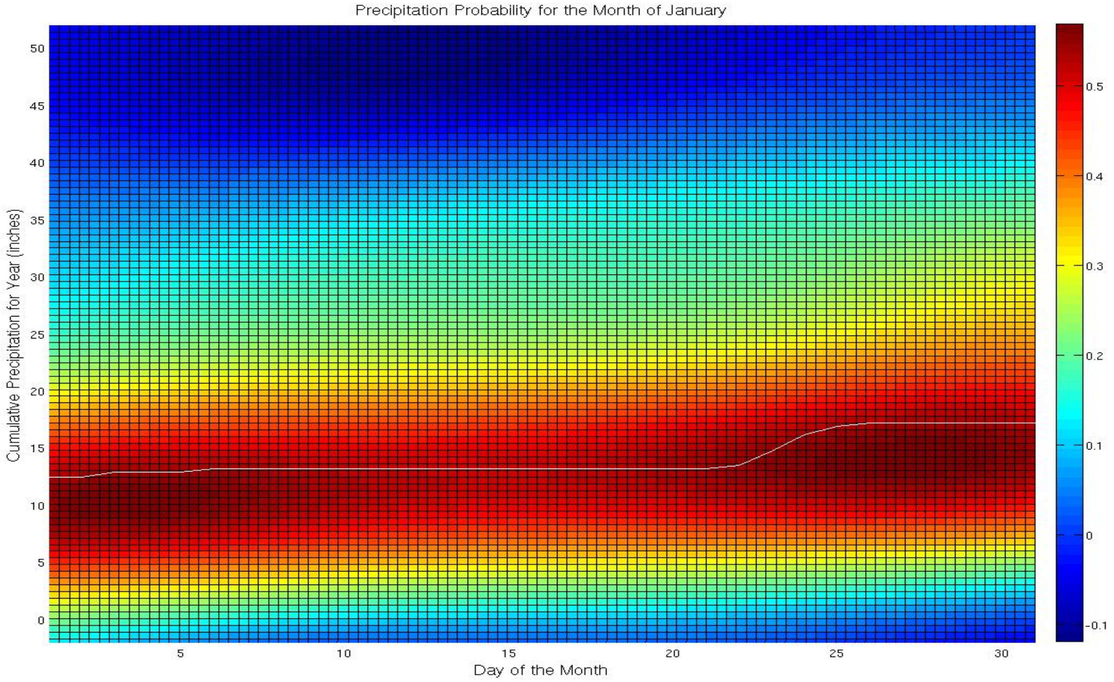

Figure 3.31: Probabilities for cumulative precipitation for the month of January at a weather station in California. The white line shows precipitation for the most recent year, which falls within the high probability range.

Chapter 4

Conclusion

The toy data sets' distributions did not always appear to be modelled well using Dettinger's method, especially when they were heavily skewed or had bimodal distributions. This may have been partially due to the way that sample data was chosen, especially since real-world data gave better results in terms of representing non-unimodal distributions. In general, the GSVD did not appear to perform significantly better or worse than the SVD, although the differences between reconstructions with the GSVD and the SVD were fairly important for some of the real-world data. Using reduced versus full versions of these decompositions also gave inconclusive results: Removing noise sometimes made Dettinger's method less prone to making histograms overly bell-shaped, but it did not always create a better approximation of the original distribution's shape than Dettinger's original method did.

For some data sets, the mean, standard deviation, skewness, and kurtosis of the original distributions were estimated more closely by various versions of Dettinger's method than by the sample data. Overall, however, there was only a small difference between using just the original data, Dettinger's method with a reduced SVD or GSVD, or the original version of Dettinger's method, to create a representation of the original probability distribution. This shows that, using any version of Dettinger's method, it is possible to create a large data set which comes very close to recapturing the moments of a sample of data. As for our different versions of Dettinger's method, none of them seem consistently more reliable than any other in terms of recapturing an original distribution's moments.

Although we were not able to get Dettinger's method to consistently provide reliable distribution estimates, this does not mean it has no value whatsoever. Because of the nature of probabilities, it is impossible to have a way to assess what the actual probability distribution is for a predicted event. As the toy data sets revealed, it is possible to get samples from a distribution which do not provide a representative sample. Dettinger's method was designed for use with very small sample sizes, when a standard PDF estimate is very difficult to perform. For such samples, any approximation of probability will have a high level of uncertainty associated with it. At the least, Dettinger's method provides a means of describing a rough probability distribution from a random sample, and gives more detail than a histogram of a small sample of data. The histogram created by a set of 5,000 data points will be much smoother than one created using only a few data points, and can thus provide an approximation of a continuous probability density curve. For small data sets on which traditional statistics cannot provide significant results, such an approximation can at least provide some framework for describing a probability on a more detailed level than simply giving means and standard deviations, especially with heavily skewed probability distributions. It can also provide a method for interpolating across gaps in data, as in section 3.1.3 where a radial basis network interpolated probabilities in between data points by using the high resolution data provided by Dettinger's method. In section 3.2.2, another interpolation showed how probability intervals can be estimated from historical data using Dettinger's method. From a purely mathematical standpoint, this method also provides an interesting example of how the SVD, GSVD, and similar tools can be used to capture information about a data set.

4.1 Further Research

Further opportunities for research in this field include looking at more forms of matrix decomposition, such as the maximum noise fraction. Also, looking for a pattern in when Dettinger's method reproduces original distributions well and when it does not could help provide new, more accurate versions of his method. Comparing the confidence intervals we found with radial basis graphs to those that would be calculated using statistics would provide a good measure of how useful our radial basis graphs could be when estimating probabilities.

Appendix A

Statistical Terminology

In order to explain the justification behind many steps in the component resampling method used in this paper, it is important to have an understanding of some basic statistical terms as they are used here. For all definitions, unless otherwise specified assume a sample of n values $x_1 \dots x_n$.

Least Squares Error: Given a method for finding the approximation, x'. of a value, x, the least squares error is the smallest possible value of $(x - x')^2$.

Sample Mean: Given a sample of data, the *sample mean* is defined as

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} (x_i)$$

Sample Standard Deviation: This is a measure of how much a sample of data points are spread about the mean, and is given by

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Correlation: A correlation is a relationship between two time series. If both series are increasing in time, they are positively correlated.

The sample correlation coefficient, r, for two data sets $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, is given by

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{A.1}$$

Variance: The variance of a set of scalars is a measure of their spread. A larger variance means the values are more spread out.

Statistically, if ($x_1 \ldots x_n$) is a set of scalars with mean $\bar{x}$ then the variance of this set is this is measured as

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_n - \bar{x})^2 \tag{A.2}$$

Covariance: This is a measure of how the entries of two vectors vary relative to one another. A covariance of 0 means that the vectors are *uncorrelated.*

$$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{A.3}$$

Covariance Matrix: If $\hat{X}$ is a matrix which has had its rows mean-subtracted, the covariance matrix is given by

$$S_{m\times m} = \frac{1}{n-1}\hat{X}\hat{X}^T \tag{A.4}$$

Each entry s_{ij} for $i \neq j$ represents the **covariance** of row i and row j (which can also be seen as the covariance of data in dimension i with that in dimension j).

Each diagonal entry s_{ii} in S for $i = 1 \ldots p$ represents the **variance** of its corresponding row.

Cross Correlation Matrix: This is similar to the covariance matrix, except each entry represents a correlation (eq. A.1) rather than a covariance (eq. A.3). The only difference is that now the matrix X' has not only been mean subtracted, but each row has also been divided by the sample standard deviation, so the matrix is standardized.

$$C_{m\times m} = \frac{1}{n-1} X' X'^{T} \tag{A.5}$$

Skewness: Skewness measures the asymmetry of a probability distribution, and is given by

$$\frac{\mu_3}{\sigma^3}$$

where μ_3 is the third moment about the mean and σ is the variance (second moment). A skewness of 0 means that the probability is symmetric about the mean. A large positive value means the data is heavily skewed to the right (it has a longer tail to the right than to the left). Negative values mean the data is skewed to the left.

Kurtosis: This is a measure of how peaked or flat a distribution is, and is related to the fourth moment about the mean via

$$\frac{\mu_4}{\sigma^4}$$

where μ_4 is the fourth moment and σ is the variance (second moment). Distributions with a high kurtosis value have a distinct peak near the mean. The kurtosis of a normal distribution is 3.

Appendix B

Linear Algebra

Here is a review of some important terms when it comes to explaining the matrix algebra behind component resampling. Unless otherwise indicated, these definitions refer to a matrix with m rows and n columns and pairs of vectors with equal dimensions.

Linear Independence: Vectors $\vec{x}$ and $\vec{y}$ are linearly independent iff the only solution to $a\vec{x} + b\vec{y} = \vec{0}$ is $a = b = 0$.

Statistical Independence: If the occurrence of one event does not affect the occurrence of another, then they are said to be independent. In other words, there is no equation $y_i = f(x_i)$ which relates any entry i in data vector $\mathbf{y}$ to entry i in data vector $\mathbf{x}$.

It is possible for vectors to be linearly independent or uncorrelated but be statistically dependent. For example, let $\vec{x} = \begin{bmatrix} cos(0) \\ cos(\pi/2) \\ cos(\pi) \\ cos(3\pi/2) \\ cos(2\pi) \end{bmatrix}$ and

$\vec{y} = \begin{bmatrix} sin(0) \\ sin(\pi/2) \\ sin(\pi) \\ sin(3\pi/2) \\ sin(2\pi) \end{bmatrix}$. The correlation coefficient and covariance of $\vec{x}$ and $\vec{y}$ are 0, but we know that these vectors are related by the fact that $x^2 + y^2 = \sin^2 t + \cos^2 t = 1$, so there is a relationship between each entry in $\vec{x}$ and its corresponding entry in $\vec{y}$ and therefore the two vectors cannot actually be independent.

Orthogonal: A set of vectors $\vec{x_1}, \ldots, \vec{x_p}$ is orthogonal if $\vec{x_i} \cdot \vec{x_j} = 0$ for all $i \neq j$.

Orthonormal: A set of vectors is orthonormal if the set is orthogonal and $||\vec{x_i}|| = 1$ for all i.

Orthogonal Projection: Given a vector $\vec{u}$, any vector can be decomposed into two vectors

$$\vec{y} = \hat{y} + \vec{z}$$

where $\hat{y} = \alpha\vec{u}$ and $\vec{z} = (\vec{y} - \alpha\vec{u})$ is orthogonal to $\vec{u}$. (In 2D, this is equivalent to saying they form a right angle; in any dimension this means that $\vec{u} \cdot \vec{z} = 0$).

Eigenvectors and Eigenvalues: If $A\vec{x} = \lambda\vec{x}$, then λ is an eigenvalue of matrix A and $\vec{x}$ is its corresponding eigenvector.

Basis: The indexed vectors $\vec{b_1}, \ldots, \vec{b_p}$ form a basis for a subspace H if any vector in H can be written as a linear combination of these vectors and they form a linearly independent set.

Best Lower Dimensional Basis: Any basis for R^m is guaranteed to provide a basis for any matrix of data in $\mathbb{R}^m$. A matrix with fewer than m columns cannot span $\mathbb{R}^m$. For our purposes, we will define the "best" k-dimensional basis for an $m \times n$ matrix (where $k < m \leq n$) as an orthonormal basis which best encapsulates most of the data. This is achieved by having basis vectors which point in the directions of the greatest variance, and thus explain as much of the variance in the data as possible.

Given a data matrix $X_{m\times n}$ with a basis and coefficient matrix $B_{m\times m}$ and $S_{m\times n}$ such that

$$X_{m\times n} = B_{m\times m}S_{m\times n}$$

there are bases $D_{m\times k}$, $k < m$, and coefficient matrices $Y_{k\times n}$ such that

$$X_{m\times n} \approx D_{m\times k}Y_{k\times n} \tag{B.1}$$

If D is the basis which can approximate X with the smallest sum of squared errors (as defined in section 2.1.1), then D is the best k-dimensional basis for X.

Row Space: The row space of a matrix $A_{m\times n}$, denoted as Row A, is the set of all linear combinations of the rows of A. Row A is a subspace of $\mathbb{R}^n$.

Column Space: The column space of a matrix $A_{m\times n}$, denoted as Col A, is the set of all linear combinations of the columns of A. Col A is a subspace of $\mathbb{R}^m$.

Null Space: The null space of a matrix $A_{m\times n}$, denoted as Nul A, is the set of all solutions (possible $\vec{x}$ vectors) to $A\vec{x} = \vec{0}$. Nul A is a subspace of $\mathbb{R}^n$.

Bibliography

[1] California Applications Program/California Climate Change Center. “2008 Scenarios – Simulation Data and Information.” Accessed June 2009. http://meteora.ucsd.edu/cap/scen08_ data.html.

[2] Dettinger, Michael. “A Component-Resampling Approach for Estimating Probability Distributions from Small Forecast Ensembles.” Climatic Change. 76: 149-168. 2006.

[3] Dettinger, Michael. “From Climate-Change Spaghetti to Climate-Change Distributions for 21st Century California.” San Francisco Estuary and Watershed Science. Vol. 3, Iss. 1: Art. 4. March 2005.

[4] Hundley, Douglas R. “Mathematical Modeling Notes.” Accessed February 2010. http://people.whitman.edu/ hundledr/courses/M350.html.

[5] Hundley, Douglas R., Michael J. Kirby, and Markus G. Anderle. ”A Solution for Blind Signal Separation Using the Maximum Noise Fraction Approach: Algorithms and Examples.” Unpublished.

[6] Lay, David C. Linear algebra and its applications. 3rd Edition. New York: Addison Wesley, 2003.

[7] Miller, Irwin and Marylees Miller. John E. Freund’s Statistics with Applications. 7th Ed. Dehli: Pearson, 2004.

[8] United States Department of Agriculture. “Natural Resources Conservation Service: California.” Accessed April 22, 2010. http://www.ca.nrcs.usda.gov/.

[9] Van Loan, Charles, and Gene Golub. Matrix Computations. Third Ed. Baltimore: The John Hopkins University Press, 1996.

[10] Zha, Hong-yuan. “Some Properties of the Quotient Singular Value Decomposition.” Journal of Computational Mathematics. Vol. 11, No.1, 1993. 50-62.