A Preliminary MML Linear Classifier using Principal Components for Multiple Classes

Lara Kornienko[1], David W. Albrecht[1], and David L. Dowe[1]

School of Computer Science and Software Engineering,
Monash University, Clayton, Victoria, 3800, Australia
{Lara.Kornienko,David.Albrecht}@csse.monash.edu.au

Abstract. In this paper we improve on the supervised classification method developed in Kornienko et al. (2002) by the introduction of Principal Component Analysis to the inference process. We also extend the classifier from dealing with binomial problems only to multinomial problems.
The classifier uses the Minimum Message Length (MML) criterion as an objective function, where the MML criterion is a Bayesian technique representing functional complexity via the length of a two-part message. The first part of the string describes the function as a model, while the second part of the string describes the data given the model in question. The MML estimator has been shown to possess desirable statistical properties and, due to the principle's intuitive and flexible nature, has already been used successfully in many different applications.
The application to which the MML criterion has been applied in this paper is the classification of objects via a linear hyperplane, where the objects are able to come from any multi-class distribution. The inclusion of Principal Component Analysis to the original inference scheme reduces the bias present in the classifier's search technique. Such improvements lead to a method which, when compared against three commercial Support Vector Machine (SVM) classifiers on Binary data, was found to be as good as the most successful SVM tested. Furthermore, the new scheme is able to classify objects of a multiclass distribution with just one hyperplane, whereas SVMs require several hyperplanes.

Key words: Machine Learning, Knowledge discovery and data mining

1 Introduction

This paper introduces a supervised classification method, which uses a Linear classifier in conjunction with a Bayesian objective function to intrinsically infer the correct distributions of a given labelled data set. The new scheme has been named PCA-Spikey, after the nature of the search procedure employed and because it takes advantage of any biases in the spread of the data by using the Principal Components as an initial set of axes on which to begin the search for the separating hyperplane. Furthermore, rather than just discriminate between

two classes of data, as most linear classifiers do, PCA-Spikey allows for any kind of multinomial distribution either side of the hyperplane.

The Bayesian method utilised in the objective function is a message length formulation called Minimum Message Length (MML) [32, 39, 34–38, 31]. It has been described as a quantitative form of Occam's Razor [21] and has its basis in data compression and Information Theory in that it expresses some information about a data set via a two-part message (a binary string), with the first part of the message being a statement of a hypothesis about the data and the second part describing the data given that hypothesis. A search procedure is then employed to find that hypothesis which minimises that string by describing the data well without being unnecessarily complicated.

The PCA-Spikey method extends the binary linear classification method, named the Spikey method, presented by the authors in [16] and [17, Chapter 5] by introducing Principal Component Analysis to the inference process. Furthermore, the capability of dealing with multinomial distributions has also been developed.

As a benchmark, we have used an implementation in Statistical Learning Theory, namely the Support Vector Machine (SVM) [27, 3]. SVMs are a popular method for performing statistical inference as they are relatively fast and flexible. Their method for discrimination involves the solving of a quadratic optimisation problem with linear inequality constraints to define a hyperplane which optimally separates data of opposing classes. These types of classifiers are called maximal margin classifiers, where the margin is defined to be the perpendicular distance of the closest points of each class to the hyperplane itself. SVMs aim to maximise the margin, whilst simultaneously minimising the classification error.

This paper begins by giving a brief overview of the MML principle in Section 2, followed by a similarly brief overview of Linear classifiers in Section 3. We then give an overview of Principal Component Analysis in Section 4, review some of the main ideas behind the Spikey encoding scheme in Section 5, and introduce the PCA-Spikey encoding scheme in Section 6. Section 7 shows how we derived a formula of a hyperplane for use in PCA-Spikey, while Sections 8 and 9 go through some of the implementation details of the PCA-Spikey encoding schemes. Section 10 give a description of the Binomial input data used to test the implementations, while Section 11 is a discussion of the results from this data. In Section 12 we introduce multinomial data to the PCA-Spikey encoding schemes and discuss the results obtained. We end with some concluding remarks in Section 13.

2 The Minimum Message Length Principle

Minimum Message Length (MML) is a Bayesian inference technique incorporating ideas from Information Theory and aims to compress the description of a set of data from an unknown source. The description of the data is a two-part message or binary string, where the structure is typically [32, 39, 34–38, 31]:

(code for **Hypothesis**) (code for **Data**| **Hypothesis**)

where the Hypothesis is a model chosen from a (continuous or discrete) hypothesis space and the Data is encoded given the chosen Hypothesis.

MML aims to find the most economical encoding of the data by minimising the length of this two-part message by finding that hypothesis which offers the most concise description, while still containing all the information necessary to re-construct that data set. This enables the hypothesis to generalise well to new data points and still be resistant to outliers. Furthermore, MML is invariant under 1-to-1 re-parameterisation for n-dimensional estimation problems [39, 34–38, 31]. For the relationship between the works of Solomonoff, Kolmogorov and Chaitin, MML and the subsequent Minimum Description Length (MDL) work of Rissanen [23], see [35–37, 31, 6]. Furthermore, MML has been able to out-perform several rival methods, where a list of such cases includes [1, 4, 6, 5, 9, 12, 13, 21, 24–26, 28–30, 35, 40].

3 Linear Classifiers

Linear classifiers use linear functions of their inputs to separate a class or classes of data from other classes. In this paper, we have used the MML principle to obtain a Linear classifier. As a benchmark, we are using Support Vector Machines (SVM) [27, 3], which, first developed by Vapnik and Chervonenkis, are able to find a linear decision surface (called a hyperplane) by simultaneously maximising the margin (perpendicular distance) between the decision hyperplane and points in the training set, and minimising the error of misclassification.

The approach we have taken with our MML classifier can be extended to handle distributions of any kind either side of its discrimination function. In this paper we have concentrated on multinomial distributions. We believe that this gives our classifier an advantage over SVMs, in that SVMs can only handle two different classes at a time with a single hyperplane. Therefore, for the multiclass problem, if using SVMs, several hyperplanes would need to be found rather than just the one.

4 Principal Component Analysis

Principal Component Analysis is a well-known technique in multivariate statistics which is often used to infer the underlying correlation structure of some data.

Looking closer at the technique, the two main functions of performing Principal Component Analysis are 1) data reduction and 2) interpretation [15]. In this research, the second function of data interpretation has been utilised in order to gain some advantage over other classification methods in the inference process. A brief description on the method for obtaining the Principal Components is as follows.

If we are given a set of data, $\mathbf{x} = (x_1, \ldots, x_N)$ in a d-dimensional space, we can analyse the variance-covariance structure of the d variables through Principal Component Analysis. This analysis finds a set of axes through the means of

the data which are called the Principal Components (see Figure 1 for the first two Principal Components of the given data set). These represent the directions of greatest variability in the data. As a consequence of this the Principal Components are sometimes thought to provide a 'natural axes' for the data and are called the Principal Axes. These are shown in Figure 2, where the data points of the previous figure are projected into the Principal Component space.

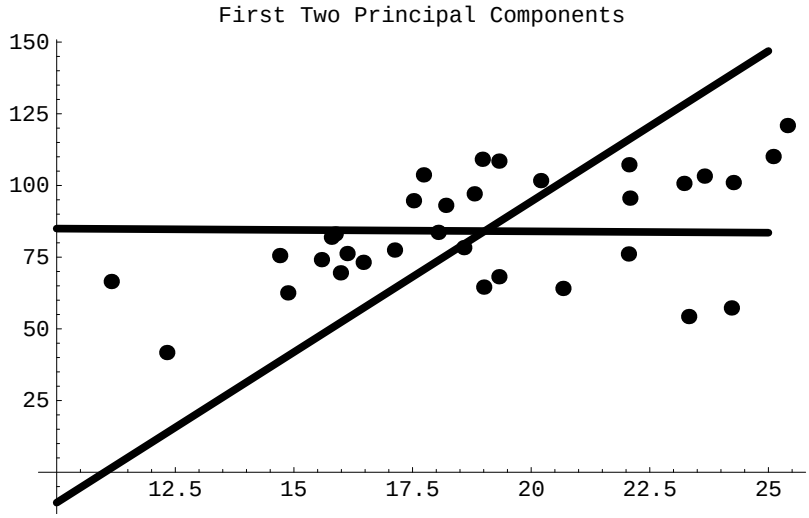

Fig. 1. Principal Component Analysis (1) for a Set of Data Points

Of course, not all of the directions found from the analysis are considered 'interesting' in the sense that if the variability of the data in a particular direction is very small, they will contribute very little to the overall variance-covariance structure of the system. Hence, it is often possible to describe the data in terms of fewer Principal Components than there are variables. Consequently, Principal Component Analysis is often used as a means of dimensionality reduction (the first function mentioned previously). Those Principal Components chosen may also reveal properties of the data that may have otherwise not been apparent (the second function mentioned previously).

To obtain the Principal Components of a given data set of size N, $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ with mean $\bar{\mathbf{x}} = 1/N \sum_{i=1}^{N} x_i$, we find the sample covariance matrix $\mathbf{S}$ for those N data points. This is defined as:

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{\mathbf{x}})(x_i - \bar{\mathbf{x}}) \tag{1}$$

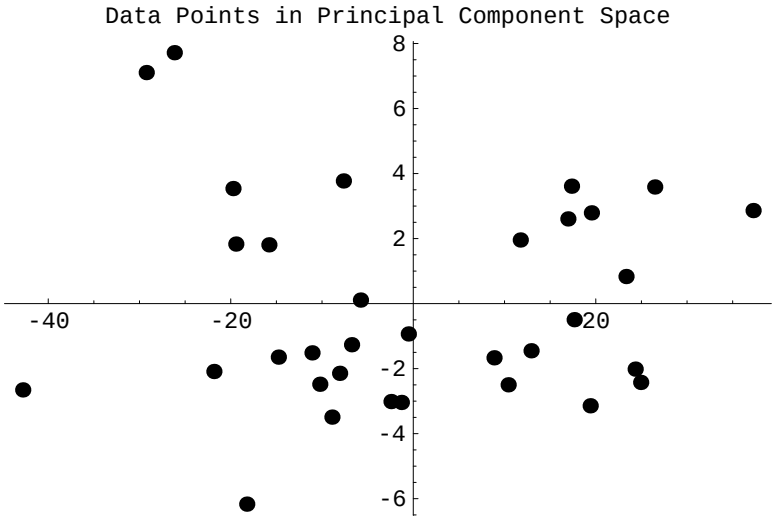

Fig. 2. Principal Component Analysis (2) for a Set of Data Points

and assume its ordered eigenvalue-eigenvector pairs are $(\lambda_1, \mathbf{a_1}), (\lambda_2, \mathbf{a_2}), \ldots, (\lambda_d, \mathbf{a_d})$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$.

The Principal Components are the linear combinations of the d variables $X_1, X_2, \ldots, X_d$:

$$\begin{aligned}
\mathbf{Y_1} &= \mathbf{a_1^T X} = a_{11}X_1 + a_{21}X_2 + \ldots + a_{d1}X_d \\
\mathbf{Y_2} &= \mathbf{a_2^T X} = a_{12}X_1 + a_{22}X_2 + \ldots + a_{d2}X_d \\
&\vdots \\
\mathbf{Y_d} &= \mathbf{a_d^T X} = a_{1d}X_1 + a_{2d}X_2 + \ldots + a_{dd}X_d
\end{aligned}$$

where each $\mathbf{a_i}$ is an eigenvector of $\mathbf{S}$.

Alternatively, the first Principal Component, $\mathbf{Y_1}$ can be found by maximising the sample variance over the coefficient vectors [20],

$$\begin{aligned}
s^2_{Y_1} &= \textstyle\sum_{i=1}^{d} \sum_{j=1}^{d} a_{i1} a_{j1} s_{ij} \\
&= \mathbf{a_1^T S a_1}
\end{aligned}$$

with the constraint that the coefficient vectors are normalised to unity, that is: $\mathbf{a_1^T a_1} = 1$.

The second Principal Component can be found in a similar way, this time maximising the variance of $\mathbf{Y_2}$ over the coefficient vectors such that

$$\mathbf{a_2^T a_2} = 1$$

but also subject to the additional constraint that the first and second coefficient vectors are orthogonal:

$$\mathbf{a_1^T a_2} = 0$$

The remaining Principal Components can also be found in a similar way, via their corresponding eigenvectors.

To find the data y_i, $i = 1, \ldots, d$, in the new coordinate space, we take the data, x_i in the original space and offset it by its mean, then pre-multiply this by the matrix $\mathbf{C}$. Where $\mathbf{C} = (\mathbf{a_1}, \ldots, \mathbf{a_d})$ is a matrix containing the normalised, orthogonal eigenvectors of the sample covariance matrix $\mathbf{S}$. That is,

$$y_i = \mathbf{C^T}(x_i - \bar{\mathbf{x}}), \qquad i = 1, \ldots, N \tag{2}$$

It can be seen from Equation 2 where the expression of Principal Component Analysis being a 'shift and rotate' comes from: the 'shift' comes from the shifting of the origin to the mean of the points and the 'rotate' comes from the multiplication by an orthogonal matrix (which, from matrix theory has a rotational effect).

5 The Spikey Codes

In [16], an initial attempt at devising a Linear classifier using MML as an objective function was achieved by expressing the physical position of the hyperplane and the encoding of the data in terms of a two-part message, whose length was then minimised in order to find the MML hyperplane. Named the 'Spikey' encoding scheme, two preliminary methods, MML_{NU}^{WT} and MML_{SR}^{WT}, arose. As an explanation of this notation, the superscript refers to the scheme used to encode the hyperplane's direction (that is, the Wallace Tree code) and the subscript refers to the scheme used to encode the hyperplane's offset (NU stands for a non-uniform code and SR refers to the Separation Ratio, which will be discussed shortly). Note that the formula for calculating the distributions on either side of the hyperplane, as described in [17] has been ammended from [16].

The general structure of the two-part MML message for the Spikey methods is,

Part I: {hyperplane's direction and offset + class distributions}
+
Part II: {class for each point}

In case of highly inseparable data, it was made an option as whether to place a hyperplane at all. For this case, the message is only made up of the cost of a single binomial distribution and the cost of the position of the hyperplane in the first part of the message is omitted.

The Hyperplane's Direction The direction of the hyperplane is defined by a perpendicular weight vector, $\mathbf{w}$, which is a unit vector passing through the origin. The co-ordinates of the tip of $\mathbf{w}$ lie on the surface (a circle in Euclidean 2-space, $\Re^2$; a sphere in Euclidean 3-space, $\Re^3$) defines the direction of $\mathbf{w}$.

To iterate over all directions, the surface was successively partitioned into regions (initially there are $2^{dimension-1}$ regions) and the midpoints of such regions used to define new directions. Please refer to [16] and [17, Section 5.2.1] for full details on how the hyperplane's direction was found.

The process of partitioning can be done an infinite number of times, each level of partition resulting in more ($dimension^{level} \times 2^{dimension-1}$) directions and smaller regions. A probability can be associated with each direction based on how many new regions are defined at that level of partitioning. This could be transformed into a message length by taking the $-\log_2$ of the probability.

The probabilities in this probability distribution sum to 1 and the sequence follows the Wallace Tree sequence [17, Section 5.2.3] [41, 16], which is based on the number of strict binary trees. The purpose of using such a sequence is to enable lower probabilities to be assigned to higher levels of partitioning.

The Hyperplane's Offset Once a hyperplane direction is defined and encoded, all data points are projected onto the line whose direction vector is $\mathbf{w}$ to find an offset. Thus, finding the offset is always reduced to one-dimension. For details, refer to [16] and [17, Section 5.2.2].

The MML_{NU}^{WT} method uses a data-independent partitioning of the unit vector $\mathbf{w}$ using a non-uniform code (similar to the discrete coding sequence for the circle), where trial hyperplanes are successively placed perpendicular to $\mathbf{w}$ such that a certain percentage of points are situated either side of the particular hyperplane being tested. The MML_{SR}^{WT} method places a hyperplane mid-way between each pair of data points using a uniform continuous prior distribution - and also takes the distance between the two current points into account by using a quantity called the Separation Ratio. Here, a hyperplane is placed midway between each pair of data points and the distance between the points is taken. This distance is then divided by the total range of the points to form a ratio, implicitly assuming a uniform prior. The $-\log_2$ of this ratio is taken to form the cost called the Separation Ratio.

The data given any particular hyperplane is encoded by calculating the distributions of points on each side of the hyperplane. Each point is encoded according to its given classification and the distribution corresponding to the side of the hyperplane it is on.

Class Distributions and Class for Each Point In the MML message, we must include some information about the distribution of points on either side of the hyperplane. This is done via the 'Class distributions' component of the first part of the message. In this case, two (or one if no hyperplane is placed) binomial distributions are inferred. Though, in general any number of any type

of distribution such as multinomial, Gaussian, Poisson, von Mises, etc. could be used.

In the second part of the message, the 'Class for each point' is the class which each point is assigned to, encoded with respect to the class distributions. This penalises misclassified points by using a longer code word to describe the position of the point than what would be used for a correctly classified point.

These two costing procedures were implemented by forming a single cost for the second part of the message. Let 'p' stand for the proportion of positive points above the hyperplane and 'q' stand for the proportion of positive points below the hyperplane. If there are N^+ positive points and N^- negative points above the hyperplane and, similarly, M^+ positive points and M^- negative points below the hyperplane, then using the method of finding the MML estimate for a binomial distribution described in [39, 16] [17, Section 3.2.2], we obtain the estimates $\widehat{p} = \frac{N^+ + 1/2}{N^+ + N^- + 1}$ and $\widehat{q} = \frac{M^+ + 1/2}{M^+ + M^- + 1}$.

Subsequently, a weighting factor of $1/2$ is added to each point and then multiplied by the $-\log_2$ of the relevant proportion. A constant, $(1 - \log_2 12)/2$, is then added for each distribution to form the overall cost as a message length [39], [38, equation (6)]:

$$
\begin{aligned}
& 2ndPartOfMessage \\
= \;& -(N^+ + 1/2)\log_2 \widehat{p} - (N^- + 1/2)\log_2(1 - \widehat{p}) + (1 - \log_2 12)/2 + \log_2 N \\
& -(M^+ + 1/2)\log_2 \widehat{q} - (M^- + 1/2)\log_2(1 - \widehat{q}) + (1 - \log_2 12)/2 + \log_2 M
\end{aligned}
$$

In the case where no hyperplane is placed, there is only a single binomial to estimate. Hence, if $\widehat{p}$ is this estimation, the cost as a message length is:

$$
\begin{aligned}
& 2ndPartOfMessage \\
= \;& -(N^+ + 1/2)\log_2 \widehat{p} - (N^- + 1/2)\log_2(1 - \widehat{p}) + (1 - \log_2 12)/2 + \log_2 N
\end{aligned}
$$

Note that some amendments to this cost presented above and in [17] were made after its publication in [16] - the terms $\log_2 N$ and $\log_2 M$ were omitted from the earlier publication, and hence were included in the ammended version present in the subsequent publications. Furthermore, although it was named the 'pq-cost' in [16], after the amendments, it's name was changed to the 'second part of the message'.

5.1 Termination Criteria for the Search Procedure

Due to the fact that this process could go on for a very long time, it was bounded in the algorithm using a test on the message length: if the cost of the current hyperplane exceeded the cost of placing no hyperplane (that is, costing the data as a single binomial), then the search was stopped. This termination criteria follows directly from the MML principle in that there is no need to continue the search if a simple model will sufficiently explain the data.

6 The PCA-Spikey Codes

PCA-Spikey is the original Spikey program with the addition of Principal Component Analysis. To further improve the original scheme, changes were made to both the encoding of the hyperplane and search heuristics. Specifically, a modification was made to the non-uniform search method for the hyperplane's offset, $\text{MML}^{\mathbf{WT}}_{\mathbf{NU}}$ in that it was broken up into two separate methods for the PCA-Spikey scheme, namely PCA-$\text{MML}^{\mathbf{WT}}_{\mathbf{NU}}$ and PCA-$\text{MML}^{\mathbf{WT}}_{\mathbf{NUR}}$. The first, PCA-$\text{MML}^{\mathbf{WT}}_{\mathbf{NU}}$ has been named the 'Quantile' method and the second, PCA-$\text{MML}^{\mathbf{WT}}_{\mathbf{NUR}}$, the 'Range' method of the non-uniform offset encoding scheme. These will be discussed shortly along with PCA-$\text{MML}^{\mathbf{WT}}_{\mathbf{SR}}$, which is a direct extension of the original Spikey scheme.

The primary aim of introducing Principal Component Analysis to the Spikey program was to improve the inference technique by obtaining a set of axes on which to perform the inference that were more representative of the natural spread of the data. Doing this would, in theory, enable inherent biases in the data to be recognised thus producing hyperplanes that were a better fit at a cheaper cost (in the MML sense). The reason for the belief that introducing Principal Component Analysis would produce cheaper hyperplanes is due to the search technique used in the Spikey scheme: hyperplanes in the direction of or perpendicular to the major axes are given the cheapest encoding costs. Thus, by transforming the original axes to the directions of greatest spread in the data, the cheapest hyperplanes are going to be amongst those which split the data perpendicular to these directions.

7 The PCA-Spikey Approach

The following is the sequence of events that occur for the PCA-Spikey approach:

1) The Principal Components are found for the skewed data, $\mathbf{x} = (x_1, \ldots, x_N)$ and the points projected into the Principal Component space. So, from Equation 2 the points in the new space are,

$$y_i = \mathbf{C^T}(x_i - \bar{\mathbf{x}}) \qquad i = 1, \ldots, N \tag{3}$$

where $\mathbf{C}$ is a matrix holding the normalised eigenvectors of the covariance matrix of the skewed data.
2) The data in the Principal Component space is then normalised by matrix $\mathbf{\Lambda}$ so that it falls within a cubed region (a square in two dimensions), where $\mathbf{\Lambda}$ is a diagonal matrix holding the reciprocal of the square roots of eigenvalues of the covariance matrix of the data $\mathbf{x}$. That is:

$$\mathbf{\Lambda} = \begin{pmatrix} 1/\sqrt{\lambda_1} & & & \\ & 1/\sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & 1/\sqrt{\lambda_d} \end{pmatrix}$$

where λ_i is the ith eigenvalue of the covariance matrix of $\mathbf{x}$.

If we call this new data $\mathbf{p}$, we can write Equation 3 as:

$$p_i = \mathbf{\Lambda C^T}(x_i - \bar{\mathbf{x}}) \qquad i = 1, \ldots, N \tag{4}$$

3) Inference is performed in the normalised Principal Component space using the Spikey search procedure (see Section 5).

Now, the general form of a hyperplane given a set of coordinates, $\mathbf{X}$ in Euclidean d-space is:

$$\mathbf{w^T X} - b \tag{5}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_d)^T$ is the weight vector associated with the hyperplane and b is the offset of the hyperplane from the origin in the direction of the weight vector.

We want an expression for the hyperplane in the uniform space given that we have a hyperplane in the normalised Principal Component space.

So, let

$$\mathbf{w_P^T P} - b_P \tag{6}$$

be a hyperplane in the normalised Principal Component space.

Then, if $\mathbf{P}$ is the coordinates in the Principal Component space normalised by $\mathbf{\Lambda}$, we can re-write Expression 6 in terms of $\mathbf{w_P}$, b, $\mathbf{X}$ and $\bar{\mathbf{x}}$:

$$\mathbf{w_P^T P} - b_P = \mathbf{w_P^T \Lambda C^T}(\mathbf{X} - \bar{\mathbf{x}}) - b_P \tag{7}$$

Re-arranging this, we get:

$$\mathbf{w_P^T P} - b_P = (\mathbf{w_P^T \Lambda C^T})\mathbf{X} - (\mathbf{w_P^T \Lambda C^T}\bar{\mathbf{x}} + b_P) \tag{8}$$

Therefore,

$$\mathbf{w^T} = \mathbf{w_P^T \Lambda C^T} \tag{9}$$

$$\mathbf{b} = \mathbf{w_P^T \Lambda C^T}\bar{\mathbf{x}} + b_P \tag{10}$$

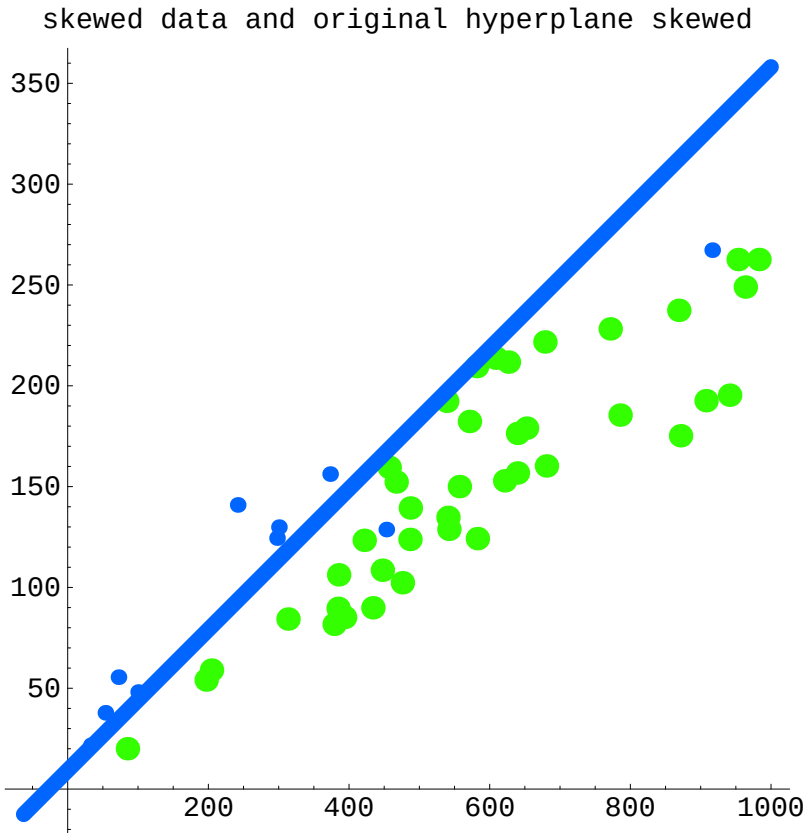

Fig. 3. Some Artificial Data Points and their Original Hyperplane

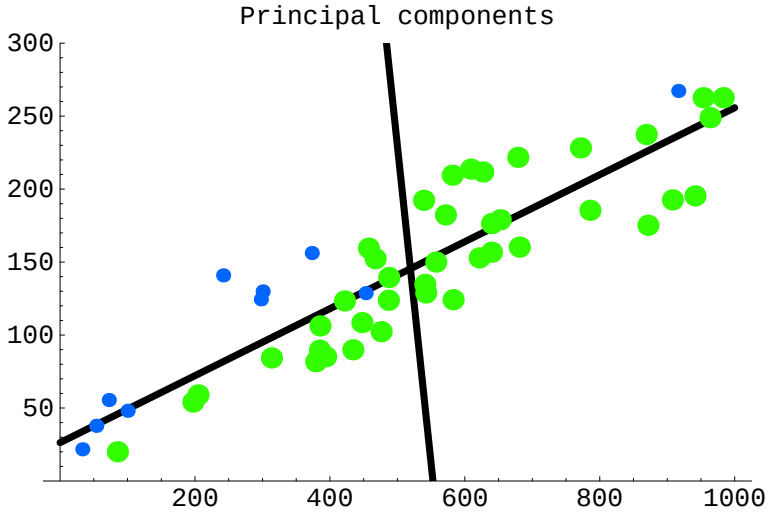

Fig. 4. Step 1 - Finding the Principal Components of a Data Set

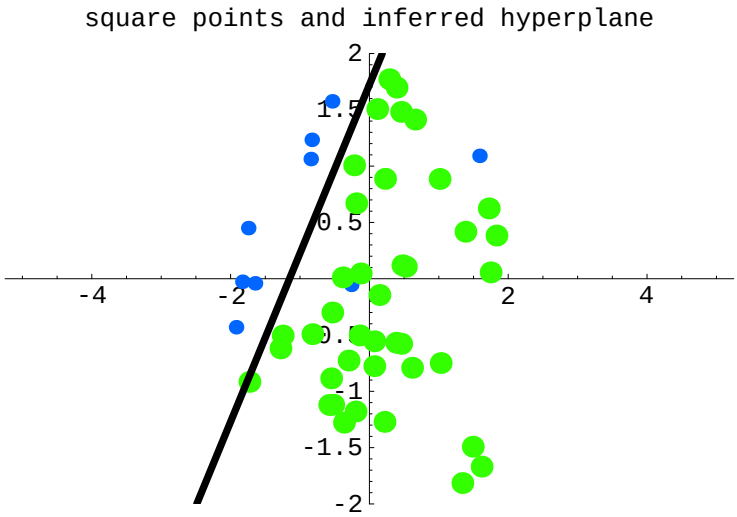

Fig. 5. Step 2 - Inferring a Hyperplane in the Normalised Principal Component Space

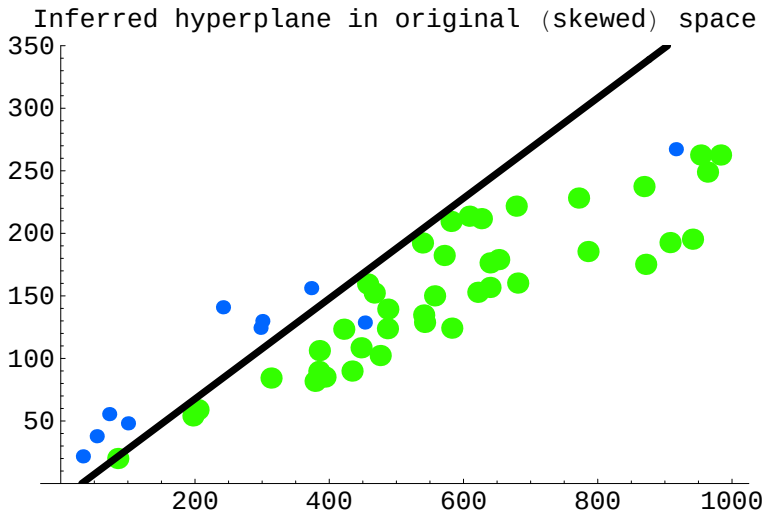

Fig. 6. Step 3 - The Inferred Hyperplane Translated Back into Original Space

8 Breaking Up the Non-Uniform Scheme for the Hyperplane's Offset: Quantile and Range Approaches

The Spikey search procedures and coding schemes for the MML hyperplane were first introduced in [16]. It was here that the non-uniform scheme, MML_{NU}^{WT} was described to find the hyperplane's offset by partitioning the data via a quantile approach. That is, a hyperplane was placed such that a certain percentage of points lay on one side of it and the rest on the other. It was decided that it might be instructive to investigate a modification to this approach, namely a scheme which defines an offset by calculating its position relative to the *range* of the data rather than calculating quantiles. That is, rather than have an offset whose position is defined by the percentage of points on either side of it, we define an offset by the proportion of the data's range it partitions. For example, the first offset position is the one which lies half way between the leftmost and rightmost data points. The next two positions are those which split the two resulting regions in half, or alternatively those which are positioned at one quarter and three quarters respectively through the range of the entire set of data points. These two methods are illustrated in Figure 7.

Although the difference between the two partitioning methods may not seem terribly significant, there is an important difference: the first one, which has been named the Quantile approach, is data dependent and therefore not invariant to changes in the position of the data points, whereas the second, the Range approach, is data independent (well, ignoring the calculation of the range) and

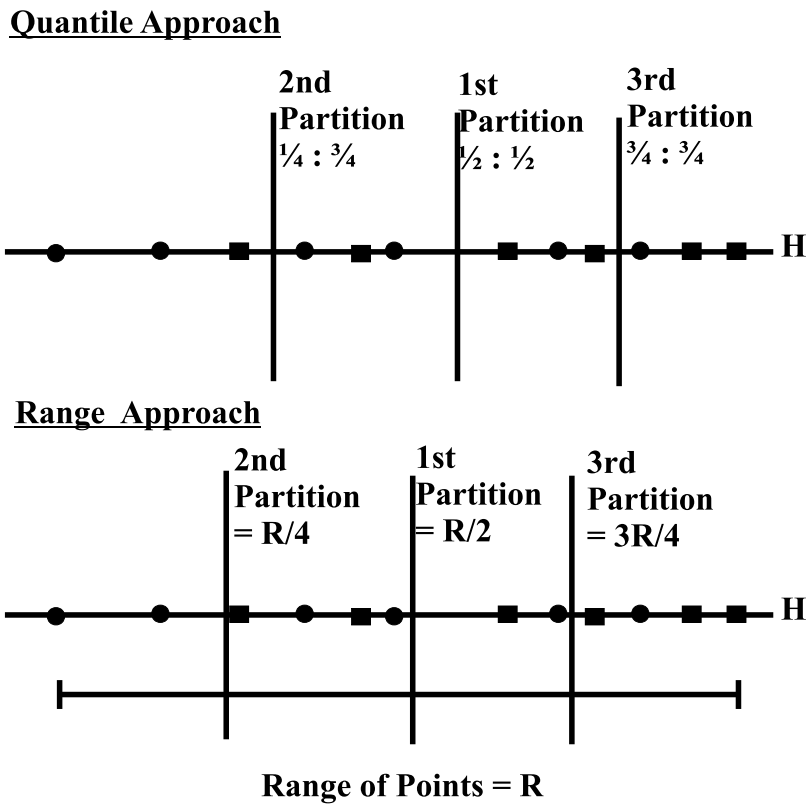

Fig. 7. The Quantile and Range Offset Search Techniques

data invariant because, if the points keep their original range, the same positions for the hyperplane's offset will be chosen if the points change their positions. It is for this reason that the second method may be the preferred one, but for the sake of investigation both were tried.

From now on, these quantile and range approaches shall be named MML_{NU}^{WT} and MML_{NUR}^{WT} respectively, the 'NU' remaining in the title to highlight the fact that they are derived from the non-uniform coding scheme for the hyperplane's offset.

9 A New Search Technique for the Hyperplane's Direction: the Angle Approach

The primary motivation for implementing a new search technique for the hyperplane's direction was the concern that the Wallace Tree coding scheme was too biased towards certain directions, and that this bias was not reflective of the spread of the data. That is, the partitioning was done relative to the coordinate axes of the space. To some extent, the introduction of Principal Component Analysis made the data more influential towards the choice of direction, but the coarseness of the Wallace Tree coding scheme still dictated the degree to which a single hyperplane could move before it took on a new cost, when in fact it seemed more sensible that the coarseness of the data should be the main factor in influencing this. For instance, it can be argued that if a hyperplane separates a set of p points correctly, the amount of uncertainty associated with that hyperplane's direction is based on the degree to which it can swing before any of those points become misclassified. Furthermore, we were aware that by using the Wallace Tree code as a prior on the hyperplane's direction we were giving the first part of the MML message perhaps an unnecessarily large weight, and the data sets composed of only a small number of points were suffering because of it. It was with these things in mind that a new approach - the angle approach, to finding the hyperplane's direction was devised. The aim of this new search method was to imitate the separation ratio search for the hyperplane's offset when looking for the hyperplane's direction. That is, we want that direction which has the largest margin of error, or can swing the most before changing its current classification. It also is different from the original Spikey and PCA-Spikey methods in that a uniform distribution is being used for the hyperplane's direction, but in two-dimensions, each direction is being specified to a precision of:

$$\frac{\text{Angle in which the hyperplane can swing}}{180^\circ}$$

noting that the normal to a hyperplane can be reflected into the upper quadrants of a unit circle.

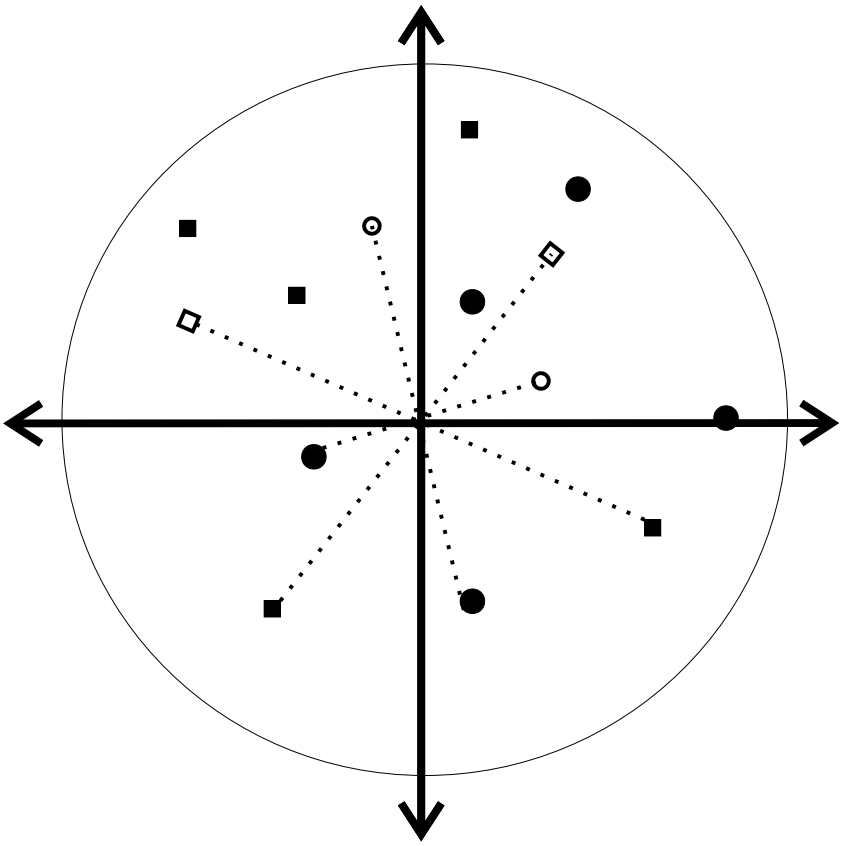

Fig. 8. Step 1 of Angle Search Procedure - Project Points in Lower Hemisphere to Upper Hemisphere

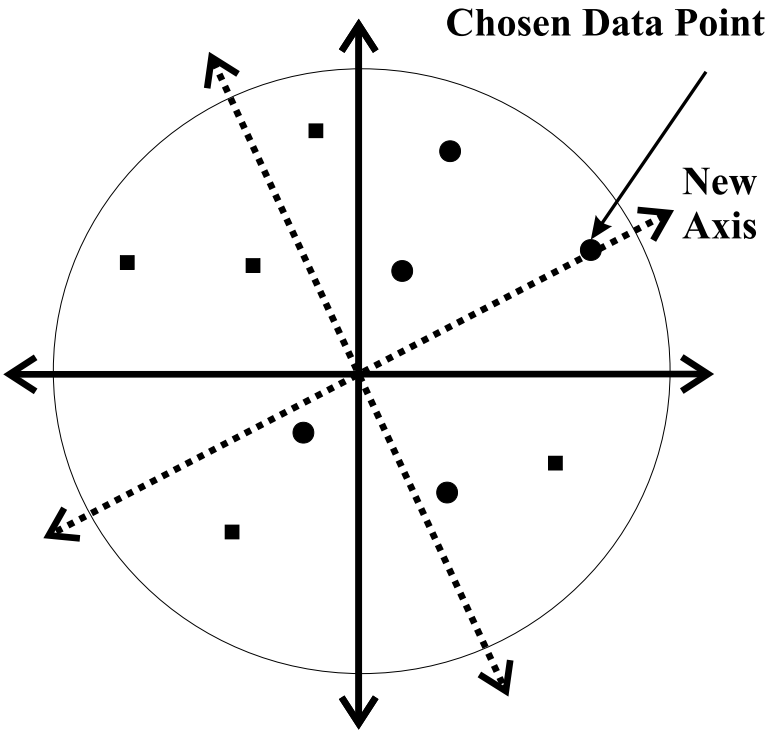

Fig. 9. Step 2 of Angle Search Procedure - Choosing the New Axis for the Angle Approach

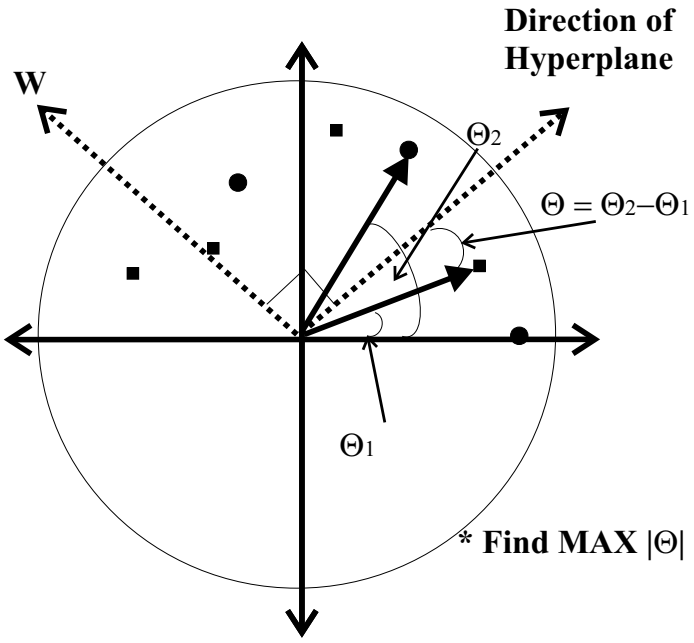

Fig. 10. Step 3 of Angle Search Procedure - The Angle Approach to Finding a Hyperplane Direction

To begin the new scheme, the Principal Component axes of the data points was calculated and all points moved into the Principal Component space and then normalised. Immediately we now had a set of points which were uniformly distributed throughout the space with the origin of the axes as the centre of the cluster of points. From here, the points were imagined to lie within a spherical region with radius equal to the distance of the furthest point from the axis origin. If we bring this down to two-dimensions only, we have the points enclosed in a circle which is broken up into four quadrants by the Principal Component axes. Since the normal to a linear hyperplane can be viewed in the top hemisphere of the circle only, it was decided that all data points in the third quadrant could be reflected into the first quadrant and all data points in the fourth quadrant could be reflected into the second quadrant (see Figure 9).

Now, an arbitrary data point was chosen and a new horizontal was defined by drawing a straight line segment from this point to the origin, thus a new set of axes can be defined by rotating the old set (see Figure 8). The angle of each data point from this new horizontal (that is, the angles will range from 0 degrees to 180 degrees) are taken and stored so that once the complete set has been obtained, they can be sorted from smallest to largest. For instance, the data point closest to the horizontal in the first quadrant will have the smallest angle while the data point closest to the horizontal in the second quadrant will have the largest angle. Since a new space is defined when the original axes are

rotated, once a hyperplane is found, all points and the inferred hyperplane must rotated back to the space defined by the original axes.

Once a list of the angles of the points from the horizontal is sorted in increasing order, we build another list having elements consisting of the difference between each pair of adjacent angles and the corresponding angle which lies mid-way between them (refer also to Figure 10). That is for m data points,

$$\text{list} = \{\{(angle_2 - angle_1), \frac{(angle_2 - angle_1)}{2} + angle_1\}, \ldots,$$
$$\{\{(angle_m - angle_{m-1}), \frac{(angle_m - angle_{m-1})}{2} + angle_{m-1}\}\}$$

This list is also sorted, but in order of decreasing difference magnitude. That is, the angle differences corresponding to data point pairs having a comparatively large gap are stored closest to the head of the list.

Now we are in a position to select the first M elements from the list, where M is a chosen integer, and extract the midpoint angles to define the directions of the hyperplanes we are going to use in our search. Note that the hyperplane itself is always defined in terms of a perpendicular weight vector, so an equivalent weight vector direction must be given. To find the weight vector, the hyperplane angle is swung 90 degrees around to either the first or second quadrant (depending on which quadrant it began in) and the equivalent x-y-coordinates are calculated via:

$$x = \cos(\theta), \quad y = \sin(\theta)$$

such that the weight vector, $\mathbf{w} = (x, y)^T$.

This direction is hence input into the PCA-Spikey program and the hyperplane's offset calculated as before.

Figure 11 shows two plots, the first being the rate of increase of the Wallace Tree code as used as a prior for the Spikey and the PCA-Spikey coding schemes as the number of iterations through the search for a hyperplane direction increases. This corresponds to an increase in the 'partition level' of the search space. The function corresponding to this plot is negative logarithm (to the base 2) of the Wallace Tree probability function, which is, for partition level m,

$$f(m) = -\log_2(\frac{(2m)!}{m!(m+1)!} \times 2^{-2m-1})$$

It can be seen from this graph that the message length increases sharply for the first 50 or so partitions, then follows a gradual upward climb from then on.

The second graph shows the rate of increase of the prior message length as for the Angle approach. Here, the negative logarithm (to the base 2) of the angle, a as a proportion of the total range of possible angles is taken, that is,

$$f(a) = -\log_2(\frac{a}{\pi})$$

$$\text{where} \quad 0 < a \leq \pi$$

It can be seen that, as one would expect from an MML formulation, the message length decreases logarithmically as the size of the angle increases. That is, the Angle method prefers to place directions in wide open regions.

Although the two graphs cannot be compared in a one to one fashion, the purpose of having them side by side was more for the benefit of discussion. If one was to compare the behaviours of the two curves, we see for the Wallace Tree code that an increase in the message length of the prior is unavoidable in the search for a hyperplane offset. The more we search, the longer the first part of the message will be. Although justifiable in the sense that higher partition levels represent greater degrees of precision for the hyperplane direction because the partitions become smaller, the rate of increase of the code was somewhat concerning. Hence, the introduction of the Angle approach.

10 The Binomial Data

10.1 Artificial Binomial Data

The first implementation of the Spikey code [16] was tested on artificial data sets which were uniformly distributed about a two-dimensional square Euclidean space. However, the introduction of Principal Component Analysis to the Spikey code was intended to discover any biases in the spread of the data. Hence, new data sets which contained some kind of inherent bias were generated to test PCA-Spikey. The bias was implemented by pre-multiplying the uniform data by the transpose of a linear translation matrix. Two such matrices were used in total, they were:

$$\text{Translation Matrix 1} = \begin{pmatrix} 10 & 0 \\ 0 & 0.25 \end{pmatrix}$$

$$\text{Translation Matrix 2} = \begin{pmatrix} 10 & 2 \\ 0.2 & 1 \end{pmatrix}$$

The new data sets have the same distribution of positive and negative points on either side and, obviously, the same number of points as their uniform counterparts. The translation matrices where chosen due to the effect they had on the uniform data. A translation which offered little or no difference from the uniform data was not chosen because the results obtained from such a set of data would most likely resemble those presented in the author's first paper [16]. Data which was skewed in some significant way would be of the most interest. Consequently, the first translation matrix (TM1) was chosen as it 'squashed' the data down and across the horizontal axis. See Figures 12 and 13 for an idea of how the uniform data (a 100 point set with a 60% / 40% distribution) is transformed by this matrix. Note that the hyperplanes have also been translated accordingly. Specifically, to define the first hyperplane, Hyperplane 1 or, $y = 1.6x + 10$ in the translated space, we take two points on this line segment, such as $(-6.25, 0)$ and $(0, 10)$ and multiply each by Translation Matrix 1, obtaining two new points:

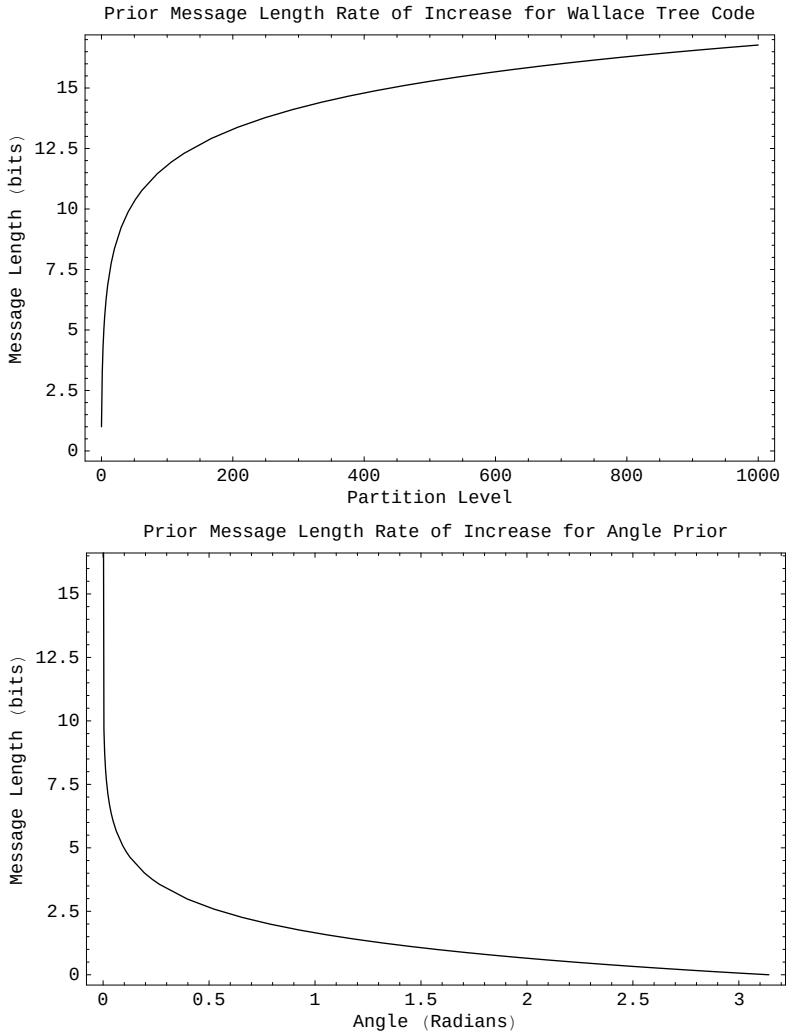

Fig. 11. Rate of Increase in Priors for PCASpikey and AngleSpikey

$$(-6.25, 0)\begin{pmatrix} 10 & 0 \\ 0 & 0.25 \end{pmatrix} = (-62.5, 0)$$

and

$$(0, 10)\begin{pmatrix} 10 & 0 \\ 0 & 0.25 \end{pmatrix} = (0, 2.5)$$

These points can then be used to find the form of the true hyperplane in the new space, that is, $y = 0.004x + 2.5$. Similarly, to find what Hyperplane 2, or $y = -0.9x + 130$ is in the translated space, we take the points on this hyperplane, $(144.44, 0)$ and $(0, 130.0)$ and multiply each by Translation Matrix 1 to obtain the new points, $(1444.4, 0)$ and $(0, 32.5)$. These give us the hyperplane, $y = -0.0225x + 32.5$. Finally, for Hyperplane 3, or $y = -1.7x + 80$, we use the points $(47.06, 0)$ and $(0, 80.0)$. Multiplying these by Translation Matrix 1 gives us the points $(470.6, 0)$ and $(0, 20.0)$, which correspond to the hyperplane $y = -0.0425x + 20$ in the translated system.

A second translation matrix, Translation Matrix 2, was also chosen to skew the uniform data. This matrix provided an even greater distortion of the original sets. Specifically, it translates a point (x, y) relative to the lines $x + 0.2y$ and $2x + y$. To see what the effect of transforming a uniform data set (the 100 point 60% / 40% distribution) using the second translation matrix, see Figures 12 and 14. Note again that the hyperplanes have also been translated accordingly. That is, using the same technique as just described, the hyperplane, $y = 1.6x + 10$ becomes $y = 0.3488x + 9.3024$. The hyperplane, $y = -0.9x + 130$ becomes $y = 0.1120x + 127.088$ and the last hyperplane, $y = -1.7x + 80$ becomes $y = 0.0311x + 79.503$ in the translated space.

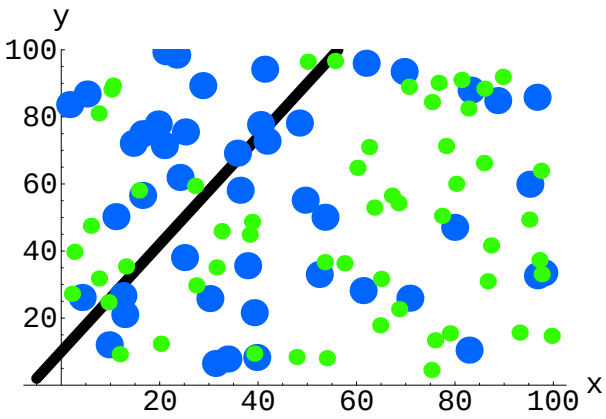

Fig. 12. Artificial Uniform Data (100pt6040) and Hyperplane 1

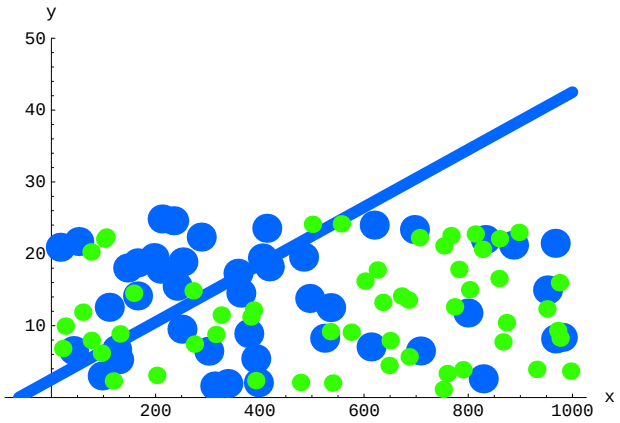

Fig. 13. Translated Artificial Data (100pt6040) and Hyperplane using TM1 and Hyperplane 1

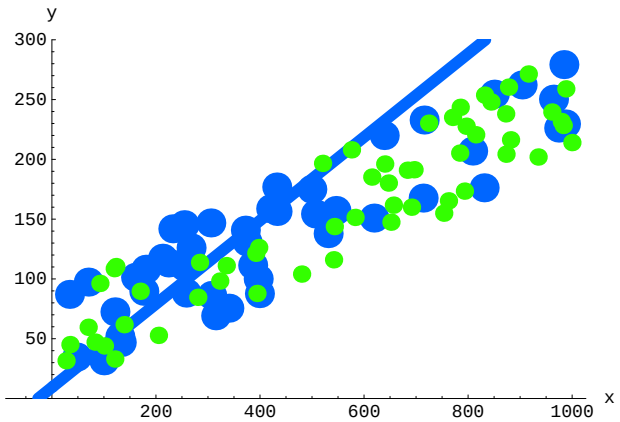

Fig. 14. Translated Artificial Data (100pt6040) and Hyperplane using TM2 and Hyperplane 1

10.2 Real Binomial Data

The real data used on the PCA-Spikey implementations was the Wisconsin Prognostic Breast Cancer Database, January 8, 1991 [2], which was also used for the original Spikey code. Similarly, the results were assessed using Probabilistic scoring [7, 21, 8, 24] [25, Section 5.1] [26, Section 3.1] [5, Section 11.4.2] and the Right/Wrong Prediction error [7] with 10-fold cross validation.

The way in which Probabilistic Scoring and the Right/Wrong Prediction error was implemented for this real data set for the PCA-Spikey methods was exactly the same as described in [17, Section 5.4.2] for the original Spikey methods, so please refer to this for more details. This data set was only run on three of the four PCA-Spikey methods, with the omission of the PCA-MML^{ANG} method, which has not yet been implemented for dimensions greater than two.

11 Results and Discussion for Binomial Data

All the PCA-Spikey methods were compared to a true hyperplane, if known, using the Kullback-Leibler distance [18] and the area between it and the true hyperplane. When no hyperplane is inferred, the Kullback-Leibler distance is taken with respect to a single binomial distribution, while the area score is not reported at all. Instead, a count of how many times this occurred is reported. If the true hyperplane was not known (as for real data), using 10-fold cross-validation in conjunction with Probabilistic Scoring [41, 8, 7, 21, 24] [25, Section 5.1] [26, Section 3.1] [5, Section 11.4.2] and the Right/Wrong Predictive Accuracy scoring metric [7].

It must be noted that whilst running SVM^{Light} on the data, on several occasions the implementation did not converge, that is, it appeared to be in an infinite loop. Hence, a time limit was put on this implementation and when any time convergence was not achieved, it was noted. This was taken into account in the calculation of the means and standard deviations of the Kullback-Leibler distance and area measures by only averaging over the number of converged cases. Consequently, some bias may be present in the results.

All results are reported at a 95% significance level using Student's t distribution on the population mean, as described in [17, Section 4.6].

11.1 General Trends for the Artificial Binomial Data

To get an overall idea of the general trends in the Kullback-Leibler distances and areas between the true and inferred hyperplanes, we refer to Tables 1 to 12. These show the means plus or minus the standard deviations of the Kullback-Leibler distances and area respectively between true and inferred hyperplanes. When discussing the results, we will break up the discussion into groups based on which Translation Matrix was used to skew the data.

In regards to the scoring techniques, it was felt that the Kullback-Leibler distances gave a better overall description of how the methods performed on

the artificial data sets, as the area scores were sometimes inconsistent (or non-existent) when the MML methods opted to not place any hyperplane a large number of times. However, when hyperplanes were placed most of the time, the two scores tended to reflect each other.

Results for Translation Matrix 1 The effect of translating the original uniform data by Translation Matrix 1 was to squash it down vertically and stretch it across horizontally. The results for the Kullback-Leibler distances for Translation Matrix 1 can be seen in Tables 1 to 6, and the results for the areas between the true and inferred hyperplanes for Translation Matrix 1 can be found in Tables 13 to 18.

For the 60% / 40% data sets, the MML methods would often not place any hyperplane, especially for the smaller data sets, instead opting to declare that the data came from one binomial distribution. This is a reasonable enough assumption as the class distributions on either side of the hyperplanes are quite close. However, it meant that the costs for the areas between the true and inferred hyperplanes were somewhat unreliable. Generally, the number of "no hyperplanes" placed and the area between the true and inferred hyperplane decreased as the data set size increased. Similarly, the Kullback-Leibler distances decreased with increasing data set size, indicating that all methods benefited from more data when the distributions either side of the true hyperplane were close. There did not appear to be any method that performed significantly better than the rest on the small data sets (the 10-point and 50-point data sets), though the MML_{SR}^{WT} method produced results which were significantly better than the others for the 10-point data sets for Hyperplane 2 and Hyperplane 3. On the larger data sets (100-points through to 1000-points) the Lagrangian proved to be the significantly better performer frequently, but the PCA-MML^{ANG} method was significantly better on the 1000-point data sets for Hyperplane 1, while for Hyperplane 2, the MML_{SR}^{WT} method performed significantly best on the 100-point data sets and the PCA-MML_{SR}^{WT} method was best on the 1000-point data sets. Another thing to note is that for the smaller data sets, there was more variation between the scores of inferred hyperplanes within each method than for the larger data sets, that is, the standard deviations tended to decrease with increasing data set size.

The 85% / 55% data sets presented a slightly easier classification problem, though still proved to be challenging. Again, there was no clear overall best performer on the smaller data sets. SMOBR was significantly best on the 10 and 50-point data sets for Hyperplane 1, whereas the MML_{SR}^{WT} method was significantly best on these for the Hyperplane 2 data sets and MML_{NU}^{WT} best for these on the Hyperplane 3 data sets. Generally, the quality of the inferences improved as the data set size increased. For most of the MML methods, there was a lower rate of "no hyperplanes" placed in comparison to the 60% / 40% data sets due to the easier distributions. Interestingly, where the PCA-Spikey methods tended to perform similarly to the SVMs on the smaller data sets, they improved on the 500 and 1000-point data sets. For the Hyperplane 1 and 3 data sets, the PCA-MML^{ANG} method performed best on these and also on the

500-point data sets for Hyperplane 2. The PCA-MML$_{SR}^{WT}$ method produced the lowest scores on the 1000-point data sets for Hyperplane 2. This clearly indicated that there is scope for improvement for the preliminary scheme presented in this thesis on small data sets.

In theory, the 95% / 05% data sets were the easiest of them all, and this was reflected in the results. In all the MML methods, except MML$_{SR}^{WT}$, hyperplanes were placed 100% of the time for data sets of size greater than 10. There was no method that was significantly better for the small data sets, with the Lagrangian SVM doing significantly well on the 10-point data sets, while the PCA-MML^{ANG} method and to a lesser extent the PCA-MML$_{SR}^{WT}$ both performing significantly better than the other methods on the 50-point data sets. As the data set size increased, however, improvements were more obvious in the MML methods than the SVM methods. The PCA-MML$_{SR}^{WT}$ was significantly better than the other methods for data sets of size greater than 50-points for all hyperplanes. Of the SVMs, however, the Lagrangian proved to be the best performer according to the significance tests. Thus, acknowledging the fact that some of the PCA-Spikey methods can provide significantly better scores than the SVM methods on easier data sets such as the 95% / 05% data sets, they still can be improved on the more difficult ones, particularly when the number of data points is low.

The general trend for the 50% / 50% data sets was that, appropriately, the MML methods would often not place a hyperplane. Consequently, the area scores did not convey any sensible information about the inferences. In regards to the Kullback-Leibler distances, in general, the MML methods tended to do significantly better than the SVM methods as the data set size increased from 50 to 1000, while SMOBR and SVM^{light} only occasionally outperformed the MML methods on the small to medium data sets (10, 50 and 100 points). Of the MML methods, the PCA-Spikey methods were significantly better than the original Spikey ones, though the MML$_{SR}^{WT}$ method did perform very well, which was in contrast to the poor behaviour of the MML$_{NU}^{WT}$ method.

To summarise the results for the artificial binomial data skewed by Translation Matrix 1, there was no overall best method, though the PCA-Spikey methods occasionally performed significantly better than the SVM methods on larger data sets and were also quite good on data which had a clear separation. The PCA-Spikey methods also performed well on the 50% / 50% data sets due to the option of placing no hyperplane. Of the MML methods, the PCA-Spikey implementations performed the best according to the significance tests as the original Spikey methods found the skewed data difficult to deal with. In particular, although the MML$_{SR}^{WT}$ method appeared to be performing quite well, it tended not to place any hyperplane when one should have been placed. The Lagrangian SVM followed by SMOBR were the significantly better performers of the tried SVMs. It does not appear that the choice of true hyperplane had a great effect on the quality of prediction indicating that the tried methods could work on a range of problems.

Results for Translation Matrix 2 The results for the Kullback-Leibler distances for Translation Matrix 2 can be seen in Tables 7 to 12 and the results for the areas between the true and inferred hyperplane can be found in Tables 19 to 24.

The effect of translating the original uniform data by Translation Matrix 2 was to stretch it upwards in a diagonal direction. Like for Translation Matrix 1, all the MML and SVM classifiers performed similarly on the translated data sets, such that again, there was no overall outstanding method, but MML_{SR}^{NU} kept refusing to place a hyperplane when one should have been placed. The choice of true hyperplane did not have a great effect on the quality of prediction, but it was noticed that there was more variation in the results on these data sets than for the Translation Matrix 1 results. This is probably because the data has an unusual spread due to the chosen Translation Matrix.

On the 60% / 40% data sets, there was a large number of 'no hyperplanes' being placed for the smaller data sets, but these decreased as the data set size increased. All implementations, both MML and SVM, showed to be performing quite well as improvements were fairly consistent as the data set size increased, but the Lagrangian SVM and the PCA-Spikey methods were the strongest performers overall according to the significance tests. On the smaller data sets (10 and 50 points), there was a fair amount of variation in the results. For instance, for Hyperplane 1, the PCA-MML^{ANG}, PCA-MML_{SR}^{WT}, MML_{SR}^{WT} methods and SMOBR all performed significantly well on the 10-point data sets, while only the PCA-MML^{ANG} method and SMOBR were significantly better than the other methods on the 50-point data sets. For Hyperplane 2, the MML_{SR}^{WT} method produced scores that showed it was significantly better than the remaining methods for the 10-point data sets and together with the PCA-MML^{ANG} method for the 50-point data sets. The results for Hyperplane 3 showed a bit more consistency, however, with the PCA-MML^{ANG}, PCA-MML_{SR}^{WT}, MML_{SR}^{WT} methods giving scores for the 10 and 50-point data sets which were significantly better than the other methods. Turning to the larger data sets (100, 500 and 1000 data points), the Lagrangian SVM and SMOBR gave the best scores on the 100-point data sets for Hyperplane 1, while the Lagrangian was the best according to the significance tests for the 500-point data sets for all hyperplanes, and the PCA-MML^{ANG} method for the 1000-point data sets for Hyperplane 1. For Hyperplane 2, the two MML Separation Ratio methods and the Lagrangian SVM performed significantly well on the 100-point data sets, but this time the PCA-MML_{SR}^{WT} method proved to be the best on the 1000-point data sets. All methods, apart from the MML_{NU}^{WT} method, gave scores for which there was no significant difference on the 100-point data sets for Hyperplane 3, where the MML_{NU}^{WT} method did not perform very well. While, for the 1000-point data sets, the PCA-MML^{ANG}, PCA-MML_{NU}^{WT} and $\text{PCA-MML}_{NUR}^{WT}$ methods were all statistically within the top-scoring implementations.

The MML methods appeared to do well on the 85% / 55% data sets, especially those containing more data points. There was a high rate of 'no hyperplanes' being placed by these methods for data sets of size 10 to 100, but

these decreased to 0 for data set sizes 500 and 1000 and generally, there was a steady improvement by all methods as the data set size increased. Overall, there were inconsistencies in the performances for the small data sets, as SMOBR and the MML_{SR}^{WT} method had significantly better scores than the other methods for the 10-point data sets for Hyperplane 1, but the MML_{SR}^{WT} method performed significantly better than the others on the 10-point data sets for Hyperplane 2. All the MML methods performed well on the 10-point data sets for Hyperplane 3, but the Lagrangian SVM was the best method, statistically speaking, for the 50-point data sets for Hyperplanes 2 and 3. For the larger data sets, the MML methods and the Lagrangian SVM all performed significantly well. Specifically, the MML_{NU}^{WT} method and SMOBR gave significantly better score than the other methods on the 100-point data sets for Hyperplane 1, but the PCA-Spikey methods were significantly the better performers on the 500-point data sets for this hyperplane. Also for Hyperplane 1, the PCA-MML^{ANG} method and the PCA-MML_{SR}^{WT} method were good on the 1000-point data sets. On the remaining large data sets, the Lagrangian SVM was significantly better than the other methods for Hyperplanes 2 and 3 with the exception of the 1000-point data sets for Hyperplane 2, where the PCA-MML_{SR}^{WT} method gave scores that were significantly higher than the rest.

For the 95% / 05% data sets, most of the MML methods only chose not to place a hyperplane for the 10 point data sets only and performed strongly in general. Again, for the small data sets, there was no outstanding performer overall and the results tended to be a bit varied. The PCA-MML^{ANG} method did appear to be yielding good scores on the smaller data sets, though, where it was amongst the top scorers for the 50-point data sets for Hyperplane 1 and the 10 and 50-point data sets for Hyperplane 2 and 3. The PCA-MML_{SR}^{WT} method also performed significantly well on the small data sets, giving results which were statistically better than the other methods for all of the 50-point data sets. Less consistent were the Lagrangian SVM, which was statistically the best performer on the 10-point data sets for Hyperplane 1 and the MML_{NU}^{WT} method, which performed the best for the 10-point data sets for Hyperplane 3. On the larger data sets of size 100 to 1000, the PCA-MML_{SR}^{WT} method performed significantly better than the others for Hyperplanes 1 and 2 and similarly well to the PCA-MML^{ANG} method on the 100-point data sets for Hyperplane 3. However, for the 500 and 1000-point data sets for Hyperplane 3, it was again significantly the best overall performer. On a further note, it appeared that the SVM's rate of improvement as the data set size increased was somewhat slower than that of the PCA-Spikey methods.

On the 50% / 50% data sets, the MML methods did not choose to place a hyperplane often, thus making the scores difficult to interpret. One thing that was apparent, though, was that the Kullback-Leibler distances of the MML methods decreased at a faster rate than for the SVMs and that the scores were better for the MML methods as the data set size increased. There did seem to be a fairly varied pattern for the small data sets, with the PCA-Spikey methods all doing significantly better than the remaining methods on the 10 to 100-point

data sets for Hyperplane 1, but SMOBR was the statistically best performer for the 10-point data sets and the MML_{SR}^{WT} method statistically best for the 50 and 100-point data sets. All of the PCA-Spikey methods and the MML_{SR}^{WT} method showed no significant difference in performance on the 10-point data sets for Hyperplane 2, where they all performed statistically better than the remaining methods, but only the PCA-MML_{SR}^{WT} and the PCA-MML^{ANG} methods performed significantly better than the rest for the 50-point data sets for this hyperplane. In reference to Hyperplane 3, the PCA-MML^{ANG} and MML_{SR}^{WT} methods gave significantly better results than the other methods on the 10-point data sets, but the PCA-MML^{ANG}, PCA-MML_{NUR}^{WT}, PCA-MML_{SR}^{WT} and MML_{SR}^{WT} methods were statistically the better methods on the 50-point data sets. Turning now to the data sets of size 100 to 1000, it appears that overall, the PCA-Spikey methods were the most consistent in providing good inferences. For instance, the PCA-Spikey methods, along with MML_{SR}^{WT}, gave significantly better results for the 100-point data sets for Hyperplane 1 than the remaining methods and, again, the PCA-Spikey methods along with SMOBR were significantly the best method for the 500-point data sets for this hyperplane. However, for the 1000-point data sets, it was the PCA-MML^{ANG}, PCA-MML_{NU}^{WT} and PCA-MML_{NUR}^{WT} methods that were statistically the strongest performers. For Hyperplane 2, the PCA-Spikey methods and the MML_{SR}^{WT} method gave scores which were significantly better than the remaining methods for the 100, 500 and 1000-point data sets. For Hyperplane 3, the scores returned by PCA-MML^{ANG}, PCA-MML_{NU}^{WT} and PCA-MML_{NUR}^{WT} showed no significant difference, and were better than the remaining methods for the 100-point data sets, while for the 500 and 1000-point data sets, as for Hyperplane 2, the PCA-Spikey methods and the MML_{SR}^{WT} method gave the best scores statistically speaking.

As a summary of the results for the artificial binomial data skewed by Translation Matrix 2, there was no overall outstanding method, though the PCA-Spikey methods and Lagrangian SVM did particularly well. It appears that, as was shown by the results for the Translation Matrix 1 data, the preliminary MML schemes can be improved for small data sets, whereas they are particularly good for larger data sets. The PCA-Spikey methods performed well for all distributions presented to it, from the difficult 60% / 40% data sets to the easier 95% / 05% data sets. The option of being able to not place any hyperplane worked to their advantage in the 50% / 50% data sets where the PCA-Spikey methods and the MML_{SR}^{WT} method were particularly good.

11.2 General Trends for the Real Binomial Data

It appears from the scores in Table 37 that for the PCA-Spikey methods, the PCA-MML_{NUR}^{WT} method was able to perform slightly better than the others PCA-Spikey methods in terms of making probabilistic predictions on the data, and equally good with the PCA-MML_{SR}^{WT} method on classifying the data into 'majority' groups, as shown by the Right/Wrong Predictive Accuracy score. Although the scores were not significantly better, it is interesting that the PCA-MML_{NUR}^{WT} method was able to give even slightly better inferences, as it

is the only method not to treat the data points individually when searching for the hyperplane's offset. If you recall, it places hyperplanes based on partitions through the *range* of the data, as opposed to looking at the *number* of data points either side of the offset as is done in the PCA-MML$_{NU}^{WT}$ method (see Section 8 for a complete description of these methods). The reason why this may be of significance is that in high dimensions, the data becomes sparser and more difficult to deal with, hence it may be useful to explore methods which are data independent when dealing with high-dimensional spaces.

When compared over the entire set of implementations run on this data - the original Spikey methods and the SVMs, as reported in Table 37, it appears that the PCA-Spikey results are somewhere in middle-range, though the PCA-MML$_{NUR}^{WT}$ method gives the best results overall, though not significantly. The reason why the PCA-Spikey methods did not produce much of an improvement over the original Spikey methods is that the data may have been reasonably uniformly spread, with no obvious skews. Consequently, the Principal Component axes would be virtually the same as the axes defined by the original Spikey methods.

12 Multinomial Data

In this section, we show that the PCA-Spikey methods can be extended to deal with multinomial data. Specifically, we test the methods on a series of artificially generated two-dimensional trinomial data sets and the well-known Iris data set [11] in both two and four dimensions. Importantly, the extensions we have implemented for multinomial data require only a single hyperplane, which is in contrast to the majority of SVM implmentations for data of this type (multi-class SVMs are discussed in [17, Chapter 10]). That is, we are trying to infer two multinomial distributions either side of the true hyperplane rather than isolate classes entirely using multiple hyperplanes. Consequently, we were unable to compare the results of the PCA-Spikey methods against any SVMs in the given time frame. We begin by introducing the artificial data sets and the evaluation method used to score the inference, we also derive the MML estimator for a trinomial distribution. The real data sets are then introduced, followed by a presentation and discussion of the entire set of results for the artificial and real data sets. It was found that on the artificial data sets, the PCA-Spikey methods tended to not place any hyperplane sometimes over 70% of the time, depending on the distribution of the data. On the Iris data sets, the methods were able to separate one of the classes from the remaining (inseparable) two on the two-dimensional data a number of times, but did not perform as well on the four-dimensional data sets.

12.1 PCA-Spikey for the Trinomial Distribution

The PCA-Spikey methods were extended from the binomial case to instead infer two trinomial distributions either side of the true hyperplane. This is different

to the approach used by SVMs in that the methods they use, such as 'One Verses the Rest' and 'Pairwise' classification require n hyperplanes for n-class classification. Any ties are then solved via error rates and voting strategies, which may not extend easily from one multiclass problem to another. The MML approach, however, is probabilistic, and with this comes the advantage that an MML classifier can be easily extended to infer any kind of distribution either side of the hyperplane by simply modifying the MML estimator used.

In order to test the PCA-Spikey methods on the trinomial data sets, expressions had to be derived for the MML estimator to cater for the trinomial distribution. Firstly, from [31] [17, Section 3.2.2], the MML estimates for a binomial distribution are given by,

$$\hat{p}_1 = \frac{n_1 + 1/2}{n_1 + n_2 + 1} \text{ and } \hat{p}_2 = \frac{n_2 + 1/2}{n_1 + n_2 + 1}$$

for a binomial with n_1 instances of class 1 and n_2 instances of class 2. Similarly, for a trinomial distribution, from the general expression given in [38, Section 2.2] and [17, Section 3.2.3], the MML estimates for a trinomial distribution are,

$$\hat{p}_1 = \frac{n_1 + 1/2}{n_1 + n_2 + n_3 + 3/2}, \ \hat{p}_2 = \frac{n_2 + 1/2}{n_1 + n_2 + n_3 + 3/2} \text{ and } \hat{p}_3 = \frac{n_3 + 1/2}{n_1 + n_2 + n_3 + 3/2}$$

with n_1 instances of class 1, n_2 instances of class 2 and n_3 instances of class 3.

12.2 Artificial Trinomial Data

In order to test the PCA Spikey inference method on multinomial data, several two-dimensional artificial trinomial data sets were generated. This was done using the hyperplanes $y = 1.6x + 10.0$ and $y = -1.7x - 80.0$. For the hyperplane $y = 1.6x + 10.0$, a trinomial distribution of 60%/30%/10% on one side and 10%/70%/20% on the other was generated and then skewed using the Translation Matrix,

$$\text{Translation Matrix 2} = \begin{pmatrix} 10 & 2 \\ 0.2 & 1 \end{pmatrix}$$

Thus, to determine the true hyperplane in this skewed space, we take two points on the line segment, $y = 1.6x + 10.0$, such as $(-6.25, 0)$ and $(0, 10)$ and multiply each by Translation Matrix 2, to obtain two new points, which can then be used to find the form of the true hyperplane in the new space. For example,

$$(-6.25, 0) \begin{pmatrix} 10 & 2 \\ 0.2 & 1 \end{pmatrix} = (-62.5, -12.5)$$

and

$$(0, 10) \begin{pmatrix} 10 & 2 \\ 0.2 & 1 \end{pmatrix} = (2, 10)$$

Therefore, the true hyperplane in the new space is, $y = 0.3488x + 9.3024$. The data sets will be referred to as the 60%/30%/10%/70% data sets, as on one side of the hyperplane the distribution of points is a trinomial having 60% of the points belonging to class 1, 30% of them belonging to class 2 and the remaining 10% of the points belonging to class 3. While, on the other side of the hyperplane, 10% of the data points belong to class 1, 70% belong to class 2 and the remaining 20% of the points belong to class 3. It was expected that this would be a relatively easy inference problem as the distributions on either side of the hyperplane are quite different.

For the next hyperplane, $y = -1.7x - 80.0$, distributions of 35%/35%/30% on one side and 40%/20%/40% on the other were used and the resulting data also skewed with respect to the above translation matrix, Translation Matrix 2. Using the same method as just described to obtain the true hyperplane in the translated space, we chose the points $(47.0588, 0)$ and $(0, 80)$ and multiply each by Translation Matrix 2, thus obtaining the new points, $(470.588, 94.1176)$ and $(16, 80)$, making our new true hyperplane, $y = 0.0311x + 79.503$. These data sets will be referred to as the 35%/35%/40%/20% data sets, using the same rationale described above.

As a matter of investigation, data sets corresponding to a special case of the trinomial distribution, namely those with only two classes present, were also generated and tested. That is, for trinomial probabilities p_1, p_2 and p_3, p_3 was set to 0. 100 such trinomial data sets were generated, each having a distribution of 95% of points being of class 1 and 5% of points being of class 2 on one side of a true hyperplane and 5% of points being of class 1 and 95% of points being of class 2 on the other side of the true hyperplane. The third class - class 3 had a 0 probability associated with it on both sides of the hyperplane. Hence, this trinomial was effectively acting as a 95%/5% binomial distribution. The true hyperplane chosen for this data was $y = 1.6x + 10.0$, and the data was translated using the first translation matrix,

$$\text{Translation Matrix 1} = \begin{pmatrix} 10 & 0 \\ 0 & 0.25 \end{pmatrix}$$

Choosing the points, $(-6.25, 0)$ and $(0, 10)$ on the true hyperplane in the original space, to find the true hyperplane in the skewed space, we multiply these by Translation Matrix 1, to obtain the new points, $(-62.5, 0)$ and $(0, 2.5)$. Thus, the true hyperplane in the skewed space is $y = 0.04x + 2.5$. The data sets will be referred to as the 95%/05%/05%/95% data sets.

12.3 Results and Discussion for Artificial Trinomial Data

The trinomial data was only run on three of the PCA-Spikey methods, omitting the Angle method due to time restrictions. In theory, however, there is no reason why this method cannot be extended to multinomial data sets. No SVM methods were tested either, as discussed earlier.

All inferred hyperplanes were scored using the Kullback-Leibler distances and areas between true and inferred hyperplanes, while, if no hyperplane was

placed, the Kullback-Leibler distance for a single Trinomial was taken and the area was not reported.

For each data set type, 100 data sets having the appropriate distributions were generated using a random number generator. All results are reported at a 95% significance level using Student's t distribution on the population mean, as described in [17, Section 4.6].

When discussing the general trends for the results obtained by the PCA-Spikey methods on the artificial trinomial data, we will be referring to the Kullback-Leibler distance tables, which are Tables 25 to 30, and the area scores between the true and inferred hyperplane, which are Tables 31 to 36.

Beginning with the 95%/05%/05%/95% data sets, the general trends for the inference scores on the trinomial data sets were the same as for their binomial equivalents, in that many of the hyperplanes inferred are the same as for the binomial data sets, and hence will not be discussed further. However, all results for the Kullback-Leibler distance can be seen in Tables 25 and 26, and the area scores between the true and inferred hyperplane can be seen in Tables 31 and 32.

Turning to the 60%/30%/10%/70% data sets, the area result tables (Tables 33 and 34) show a high number of 'no hyperplanes' being inferred by all methods, reflecting the difficulty of the data sets. In general, the Kullback-Leibler distances decreased as the data set size increased (Tables 27 and 28), however, due to the lack of hyperplanes, the area scores were somewhat inconsistent. There did not appear to be any one significantly best method, nor did any one method tend to place hyperplanes any more than the others.

The same can be said for the 35%/35%/40%/20% data sets - due to a large number of 'no hyperplanes' being inferred as a result of the distributions on either side of the true hyperplane being very close, the area scores were inconsistent (Tables 35 and 36) and hence, inconclusive. However, for all methods, the Kullback-Leibler distances (Tables 29 and 30) decreased steadily as the data set size increased. There was no method that stood out as being significantly better than the others.

12.4 Real Trinomial Data

The first real data set tested was the well-known Iris data set [11], which holds classes of three different Iris plants. This data set contains 150 data points altogether, where each of the three classes contains 50 points and each point consists of four numeric attributes and a class specification of either 1, 2 or 3. The attributes, all measured in centimetres are: 1) sepal length, 2) sepal width, 3) petal length and 4) petal width. All classes have an identical distribution of 33.3%, but only one of the classes is linearly separable from the other two, which are not linearly separable from each other.

Before jumping straight into the four-dimensional problem, the data was tested on just the last two attributes, petal length and petal width. As discussed in [17, Section 5.4.2], since no 'true' hyperplane exists for real data, the same

scoring methods for the artificial data could not be used for the real data. Consequently, to obtain training and test sets for the Iris data, 10-fold cross-validation was used along with Probabilistic Scoring and the Right/Wrong Predictive Accuracy measure as test metrics.

12.5 General Trends for the Real Trinomial Data

Tables 38 and 39 shows the results for the four PCA-Spikey methods when run on the two-dimensional and four-dimensional Iris data sets [11] respectively. The first column of each table refers to the results obtained using Probabilistic Scoring and the second column refers to the Right/Wrong Predictive Accuracy score. Both scoring techniques were used in conjunction with 10-fold cross-validation. Note that the Probabilistic Scoring entry should be as small as possible for a 'good' inference and the Right/Wrong Predictive accuracy should be as large as possible.

We will begin with a discussion of the general trends of the data, treating the two-dimensional and four-dimensional results separately

Note that the PCA-Spikey methods were the only methods to be run on this data due to the fact that they are able to take trinomial data whereas the others are not.

The general trends for the Iris data are presented as follows, beginning with the Iris data in two-dimensions, then followed by the Iris data in four-dimensions.

Iris in Two-Dimensions Referring to Table 38, we see that although the PCA-MML$^{WT}_{NUR}$ method appears to have achieved the lowest average Probabilistic Scoring results and equal highest average Right/Wrong Predictive Accuracy with the PCA-MML$^{WT}_{SR}$ method on the Iris data in two-dimensions, on closer inspection, there was no significant difference between the methods overall. It had been expected that the PCA-MML$^{WT}_{NUR}$ and PCA-MML$^{WT}_{SR}$ methods may be, on average, slightly better than the PCA-MML$^{WT}_{NU}$ method due to the limitations of the offset search technique. The Non-Uniform (NU) offset search technique partitions the data such that a certain percentage of data points lie on one side of the hyperplane and the rest lie on the other side. The percentages follow the pattern such that the data is grouped into halves for the first partition, quarters for the next partitions, eighths for the next and so on. Thus, thirds never occur. For the Iris data, this works to the method's disadvantage, as the data set contains three classes with an equal number of points in each. Hence, a clear separation of one of the classes from the rest will not occur for certain directions of the hyperplane. The remaining methods do not have this problem, though the PCA-MML$^{WT}_{NUR}$ method could run into similar difficulties on other data sets.

Iris in Four-Dimensions The PCA-Spikey methods were then tested on the full Iris data set, which is four dimensional. The relevant scores are shown in Table 39.

It appears from the scores that the PCA-Spikey methods found this data set quite difficult. In particular, the Right/Wrong Predictive Accuracy scores showed that on average, the correct class was chosen only about half of the time, though it must be pointed out that the best that can be done on this data set is a Predictive Accuracy score of between 60% and 70%, and that inferences will improve given more data. Nonetheless, this obviously points out a weakness in the PCA-Spikey methods and can be an area to investigate in the future - to improve the PCA-Spikey methods on real trinomial data in dimensions greater than two. Though the PCA-MML$_{NUR}^{WT}$ method appears to have the best scores, it did not perform significantly better than the other two methods, suggesting an improvement is needed within the Spikey scheme itself.

13 Conclusion

The first main point of this paper was to show that the original Spikey encoding scheme described in [17, Chapter 5] [16] could be improved by the introduction of Principal Component Analysis by using the Principal Components as an initial set of axes on which to begin the search for the separating hyperplane. The second main point was to show that a Linear classifier obtained from the MML principle can be as effective on both artificially generated and real-world binary data as the most successful SVM tested in this study. The final point was to show that the PCA-Spikey methods were able to deal with multinomial data using a single hyperplane, where they were tested on trinomial data sets.

The PCA-Spikey methods out-performed the original Spikey methods. Also it was found that the PCA-Spikey methods performed significantly better than the SVMs on the larger artificial Binomial data sets tested, while the SVMs tended to dominate the smaller data sets. On the real Binomial data sets, due to the fact that the data was not highly skewed, the PCA-Spikey methods performed similarly to the original Spikey methods, and they were as good as the best SVM on that data. All the PCA-Spikey methods performed similarly for the artificial Trinomial data sets. The results for the real Iris data set, which was tested as both a two-dimensional and a four-dimensional problem showed that, for the two-dimensional set, the PCA-MML$_{NU}^{WT}$ method had some difficulties due to a weakness in its search procedure for the hyperplane's offset. However, the PCA-MML$_{NUR}^{WT}$ method and the PCA-MML$_{SR}^{WT}$ method were able to distinguish the separable class from the two inseparable classes each time.

The four-dimensional Iris data set proved to be quite a challenge, meaning the scores did not prove to be very encouraging. Future research may include improving the PCA-Spikey code for use in higher dimensions, or otherwise investigating a method tailored specifically for dimensions greater than 2 or 3.

Overall, several improvements can be made to the PCA-Spikey methods in terms of the search procedures and coding schemes used, particularly for smaller data sets. A possible alternative is the encoding of data points to geometrically define a hyperplane, similar to the Support Vectors in SVMs (for an alternative MML approach to a related problem using both Lung Cancer and Breast

Cancer data, which does not use Principal Components, see [26]). Furthermore, the type of distribution used for the input data may be varied to non-uniform distributions. Another recent development in MML that may be looked into has been in Comley and Dowe [5, 6], where a concrete application of Dowe's abstract notion of inverse learning [10] to generalised Bayesian networks including a mix of both continuous and discrete variables is given. These and other ideas may be investigated in the future.

PCA-Spikey Results Tables for Binomial - Kullback-Leibler Distance Between True and Inferred Hyperplanes

A Kullback-Leibler Tables for Artificial Binomial Data, Translation Matrix 1

The following tables show the Kullback-Leibler distances between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, two of the Spikey methods ([16, 17]) and the SVMs, SVM^{light}, the Lagrangian SVM and SMOBR for artificial Binary data skewed via Translation Matrix 1.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0467 ± 0.0404	0.0228 ± 0.0134	0.0201 ± 0.0114
PCA-MML$_{SR}^{WT}$	0.0418 ± 0.0405	0.0210 ± 0.0108	0.0189 ± 0.0106
PCA-MML$_{NUR}^{WT}$	0.0497 ± 0.0519	0.0227 ± 0.0131	0.0192 ± 0.0077
PCA-MML^{ANG}	0.0405 ± 0.0378	0.0196 ± 0.0033	0.0182 ± 0.0101
MML_{NU}^{WT}	0.0476 ± 0.0357	0.0240 ± 0.0150	0.0180 ± 0.0085
MML_{SR}^{WT}	0.0346 ± 0.0128	0.0196 ± 0.0033	0.0172 ± 0.0017
SVM^{light}	0.0445 ± 0.0301	0.0200 ± 0.0066	0.0167 ± 0.0040
Lagrangian	0.0654 ± 0.0569	0.0211 ± 0.0133	0.0159 ± 0.0092
SMOBR	0.0344 ± 0.0168	0.0179 ± 0.0054	0.0157 ± 0.0043
85/55			
PCA-MML$_{NU}^{WT}$	0.0607 ± 0.0693	0.0475 ± 0.0216	0.0424 ± 0.0131
PCA-MML$_{SR}^{WT}$	0.0541 ± 0.0638	0.0462 ± 0.0234	0.0422 ± 0.0141
PCA-MML$_{NUR}^{WT}$	0.0533 ± 0.0598	0.0471 ± 0.0217	0.0423 ± 0.0140
PCA-MML^{ANG}	0.0522 ± 0.0623	0.0481 ± 0.0305	0.0419 ± 0.0150
MML_{NU}^{WT}	0.0569 ± 0.0552	0.0409 ± 0.0147	0.0344 ± 0.0096
MML_{SR}^{WT}	0.0400 ± 0.0286	0.0410 ± 0.0071	0.0390 ± 0.0042
SVM^{light}	0.0558 ± 0.0433	0.0434 ± 0.0123	0.0396 ± 0.0059
Lagrangian	0.0926 ± 0.0835	0.0482 ± 0.0174	0.0422 ± 0.0089
SMOBR	0.0419 ± 0.0351	0.0370 ± 0.0105	0.0354 ± 0.0067
95/05			
PCA-MML$_{NU}^{WT}$	0.2034 ± 0.0510	0.1131 ± 0.0532	0.0814 ± 0.0446
PCA-MML$_{SR}^{WT}$	0.2025 ± 0.0523	0.0924 ± 0.0516	0.0481 ± 0.0419
PCA-MML$_{NUR}^{WT}$	0.2090 ± 0.0462	0.1086 ± 0.0536	0.0797 ± 0.0408
PCA-MML^{ANG}	0.1763 ± 0.0754	0.0661 ± 0.0479	0.0748 ± 0.0492
MML_{NU}^{WT}	0.1882 ± 0.0435	0.1593 ± 0.0871	0.0881 ± 0.0589
MML_{SR}^{WT}	0.2167 ± 0.0409	0.3652 ± 0.0208	0.3679 ± 0.0109
SVM^{light}	0.1844 ± 0.0268	0.2826 ± 0.0433	0.2549 ± 0.0274
Lagrangian	0.0545 ± 0.0412	0.0934 ± 0.0489	0.0690 ± 0.0349
SMOBR	0.1932 ± 0.0645	0.2457 ± 0.0739	0.1922 ± 0.0548
50/50			
PCA-MML$_{NU}^{WT}$	0.0108 ± 0.0356	0.0034 ± 0.0207	0.0001 ± 0.0001
PCA-MML$_{SR}^{WT}$	0.0072 ± 0.0068	0.0046 ± 0.0247	0.0001 ± 0.0001
PCA-MML$_{NUR}^{WT}$	0.0145 ± 0.0443	0.0048 ± 0.0254	0.0009 ± 0.0075
PCA-MML^{ANG}	0.0072 ± 0.0068	0.0020 ± 0.0161	0.0001 ± 0.0001
MML_{NU}^{WT}	0.0226 ± 0.0362	0.0118 ± 0.0224	0.0040 ± 0.0063
MML_{SR}^{WT}	0.0072 ± 0.0068	0.0004 ± 0.0005	0.0001 ± 0.0001
SVM^{light}	0.0311 ± 0.0344	0.0076 ± 0.0088	0.0028 ± 0.0044
Lagrangian	0.0569 ± 0.0667	0.0121 ± 0.0177	0.0067 ± 0.0084
SMOBR	0.0085 ± 0.0078	0.0025 ± 0.0068	0.0008 ± 0.0022

Table 1. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0133 ± 0.0052	0.0046 ± 0.0034
PCA-MML$_{SR}^{WT}$	0.0156 ± 0.0044	0.0068 ± 0.0046
PCA-MML$_{NUR}^{WT}$	0.0143 ± 0.0039	0.0073 ± 0.0034
PCA-MML^{ANG}	0.0158 ± 0.0021	0.0035 ± 0.0034
MML_{NU}^{WT}	0.0117 ± 0.0018	0.0096 ± 0.0029
MML_{SR}^{WT}	0.0156 ± 0.0004	0.0154 ± 0.0002
SVM^{light}	0.0140 ± 0.0027	0.0146 ± 0.0018
Lagrangian	0.0068 ± 0.0034	0.0047 ± 0.0027
SMOBR	0.0130 ± 0.0035	0.0124 ± 0.0036
85/55		
PCA-MML$_{NU}^{WT}$	0.0099 ± 0.0073	0.0052 ± 0.0038
PCA-MML$_{SR}^{WT}$	0.0109 ± 0.0078	0.0046 ± 0.0039
PCA-MML$_{NUR}^{WT}$	0.0143 ± 0.0065	0.0101 ± 0.0043
PCA-MML^{ANG}	0.0082 ± 0.0071	0.0033 ± 0.0028
MML_{NU}^{WT}	0.0185 ± 0.0116	0.0087 ± 0.0084
MML_{SR}^{WT}	0.0400 ± 0.0008	0.0401 ± 0.0004
SVM^{light}	0.0399 ± 0.0008	0.0402 ± 0.0004
Lagrangian	0.0391 ± 0.0011	0.0390 ± 0.0008
SMOBR	0.0353 ± 0.0035	0.0384 ± 0.0015
95/05		
PCA-MML$_{NU}^{WT}$	0.0352 ± 0.0233	0.0303 ± 0.0239
PCA-MML$_{SR}^{WT}$	0.0103 ± 0.0087	0.0057 ± 0.0049
PCA-MML$_{NUR}^{WT}$	0.0620 ± 0.0200	0.0683 ± 0.0136
PCA-MML^{ANG}	0.0220 ± 0.0148	0.0123 ± 0.0123
MML_{NU}^{WT}	0.0448 ± 0.0309	0.0396 ± 0.0231
MML_{SR}^{WT}	0.3883 ± 0.0019	0.3902 ± 0.0010
SVM^{light}	0.2553 ± 0.0142	0.2488 ± 0.0029
Lagrangian	0.0390 ± 0.0190	0.0295 ± 0.0156
SMOBR	0.0774 ± 0.0266	0.0927 ± 0.0395
50/50		
PCA-MML$_{NU}^{WT}$	0.0043 ± 0.0132	0.0001 ± 0.0006
PCA-MML$_{SR}^{WT}$	0.0041 ± 0.0124	1.3644e-6 ± 1.3243e-6
PCA-MML$_{NUR}^{WT}$	0.0040 ± 0.0121	0.0001 ± 0.0007
PCA-MML^{ANG}	0.0041 ± 0.0124	1.3644e-6 ± 1.3243e-6
MML_{NU}^{WT}	0.0046 ± 0.0107	0.0005 ± 0.0007
MML_{SR}^{WT}	0.0041 ± 0.0124	1.3644e-6 ± 1.3243e-6
SVM^{light}	0.0036 ± 0.0102	0.0004 ± 0.0005
Lagrangian	0.0060 ± 0.0140	0.0006 ± 0.0008
SMOBR	0.0036 ± 0.0104	0.0001 ± 0.0003

Table 2. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0544 ± 0.0686	0.0195 ± 0.0137	0.0157 ± 0.0066
PCA-MML$_{SR}^{WT}$	0.0351 ± 0.0340	0.0175 ± 0.0100	0.0162 ± 0.0144
PCA-MML$_{NUR}^{WT}$	0.0473 ± 0.0584	0.0188 ± 0.0116	0.0165 ± 0.0110
PCA-MML^{ANG}	0.0392 ± 0.0490	0.0165 ± 0.0037	0.0161 ± 0.0139
MML_{NU}^{WT}	0.0418 ± 0.0413	0.0265 ± 0.0185	0.0179 ± 0.0091
MML_{SR}^{WT}	0.0288 ± 0.0121	0.0165 ± 0.0037	0.0148 ± 0.0019
SVM^{light}	0.0478 ± 0.0431	0.0186 ± 0.0095	0.0148 ± 0.0032
Lagrangian	0.0768 ± 0.0797	0.0209 ± 0.0129	0.0135 ± 0.0076
SMOBR	0.0316 ± 0.0134	0.0187 ± 0.0088	0.0164 ± 0.0130
85/55			
PCA-MML$_{NU}^{WT}$	0.0604 ± 0.0730	0.0416 ± 0.0270	0.0343 ± 0.0082
PCA-MML$_{SR}^{WT}$	0.0559 ± 0.0726	0.0349 ± 0.0200	0.0342 ± 0.0108
PCA-MML$_{NUR}^{WT}$	0.0586 ± 0.0687	0.0391 ± 0.0205	0.0339 ± 0.0076
PCA-MML^{ANG}	0.0593 ± 0.0848	0.0335 ± 0.0133	0.0325 ± 0.0066
MML_{NU}^{WT}	0.0553 ± 0.0588	0.0416 ± 0.0244	0.0337 ± 0.0102
MML_{SR}^{WT}	0.0335 ± 0.0290	0.0325 ± 0.0067	0.0321 ± 0.0041
SVM^{light}	0.0537 ± 0.0450	0.0351 ± 0.0160	0.0322 ± 0.0042
Lagrangian	0.1013 ± 0.0910	0.0425 ± 0.0169	0.0356 ± 0.0070
SMOBR	0.0358 ± 0.0298	0.0349 ± 0.0124	0.0333 ± 0.0057
95/05			
PCA-MML$_{NU}^{WT}$	0.1730 ± 0.0503	0.0923 ± 0.0670	0.0645 ± 0.0425
PCA-MML$_{SR}^{WT}$	0.1728 ± 0.0610	0.0555 ± 0.0519	0.0371 ± 0.0282
PCA-MML$_{NUR}^{WT}$	0.1837 ± 0.0692	0.0794 ± 0.0589	0.0608 ± 0.0360
PCA-MML^{ANG}	0.1424 ± 0.0580	0.0550 ± 0.0470	0.0710 ± 0.0518
MML_{NU}^{WT}	0.1808 ± 0.0495	0.2220 ± 0.0669	0.1130 ± 0.0968
MML_{SR}^{WT}	0.1770 ± 0.0447	0.3046 ± 0.0264	0.3220 ± 0.0099
SVM^{light}	0.1779 ± 0.0497	0.2967 ± 0.0297	0.3183 ± 0.0166
Lagrangian	0.1141 ± 0.0967	0.0817 ± 0.0524	0.0618 ± 0.0350
SMOBR	0.1781 ± 0.0449	0.3046 ± 0.0265	0.3221 ± 0.0099
50/50			
PCA-MML$_{NU}^{WT}$	0.0155 ± 0.0601	0.0068 ± 0.0260	0.0006 ± 0.0050
PCA-MML$_{SR}^{WT}$	0.0050 ± 0.0060	0.0019 ± 0.0145	0.0007 ± 0.0051
PCA-MML$_{NUR}^{WT}$	0.0089 ± 0.0285	0.0056 ± 0.0233	0.0001 ± 0.0002
PCA-MML^{ANG}	0.0086 ± 0.0357	0.0021 ± 0.0165	0.0001 ± 0.0002
MML_{NU}^{WT}	0.0191 ± 0.0331	0.0107 ± 0.0235	0.0056 ± 0.0077
MML_{SR}^{WT}	0.0050 ± 0.0060	0.0004 ± 0.0005	0.0001 ± 0.0002
SVM^{light}	0.0319 ± 0.0331	0.0075 ± 0.0108	0.0043 ± 0.0050
Lagrangian	0.0529 ± 0.0704	0.0130 ± 0.0212	0.0059 ± 0.0076
SMOBR	0.0069 ± 0.0077	0.0051 ± 0.0114	0.0027 ± 0.0070

Table 3. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0122 ± 0.0038	0.0046 ± 0.0034
PCA-MML$_{SR}^{WT}$	0.0136 ± 0.0035	0.0035 ± 0.0033
PCA-MML$_{NUR}^{WT}$	0.0124 ± 0.0033	0.0057 ± 0.0040
PCA-MML^{ANG}	0.0134 ± 0.0015	0.0069 ± 0.0046
MML_{NU}^{WT}	0.0122 ± 0.0020	0.0111 ± 0.0016
MML_{SR}^{WT}	0.0133 ± 0.0004	0.0131 ± 0.0002
SVM^{light}	0.0133 ± 0.0004	0.0131 ± 0.0002
Lagrangian	0.0065 ± 0.0032	0.0051 ± 0.0028
SMOBR	0.0130 ± 0.0013	0.0126 ± 0.0010
85/55		
PCA-MML$_{NU}^{WT}$	0.0103 ± 0.0068	0.0055 ± 0.0043
PCA-MML$_{SR}^{WT}$	0.0068 ± 0.0067	0.0039 ± 0.0029
PCA-MML$_{NUR}^{WT}$	0.0096 ± 0.0079	0.0046 ±0.0041
PCA-MML^{ANG}	0.0088 ± 0.0073	0.0038 ± 0.0028
MML_{NU}^{WT}	0.0284 ± 0.0057	0.0165 ± 0.0114
MML_{SR}^{WT}	0.0335 ± 0.0007	0.0335 ± 0.0004
SVM^{light}	0.0334 ± 0.0007	0.0335 ± 0.0005
Lagrangian	0.0334 ± 0.0015	0.0331 ± 0.0008
SMOBR	0.0340 ± 0.0015	0.0338 ± 0.0008
95/05		
PCA-MML$_{NU}^{WT}$	0.0382 ± 0.0224	0.0351 ± 0.0196
PCA-MML$_{SR}^{WT}$	0.0126 ± 0.0101	0.0064 ± 0.0063
PCA-MML$_{NUR}^{WT}$	0.0305 ± 0.0184	0.0242 ± 0.0151
PCA-MML^{ANG}	0.0240 ± 0.0182	0.0147 ± 0.0149
MML_{NU}^{WT}	0.0376 ± 0.0197	0.0322 ± 0.0136
MML_{SR}^{WT}	0.3366 ± 0.0022	0.3388 ± 0.0010
SVM^{light}	0.3367 ± 0.0020	0.3379 ± 0.0078
Lagrangian	0.0350 ± 0.0192	0.0305 ± 0.0154
SMOBR	0.3366 ± 0.0022	0.3388 ± 0.0010
50/50		
PCA-MML$_{NU}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML$_{SR}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML$_{NUR}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML^{ANG}	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
MML_{NU}^{WT}	0.0015 ± 0.0019	0.0007 ± 0.0009
MML_{SR}^{WT}	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
SVM^{light}	0.0006 ± 0.0007	0.0003 ± 0.0003
Lagrangian	0.0012 ± 0.0012	0.0006 ± 0.0007
SMOBR	0.0007 ± 0.0016	0.0003 ± 0.0005

Table 4. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0486 ± 0.0589	0.0202 ± 0.0191	0.0148 ± 0.0065
PCA-MML$_{SR}^{WT}$	0.0314 ± 0.0172	0.0179 ± 0.0135	0.0150 ± 0.0090
PCA-MML$_{NUR}^{WT}$	0.0387 ± 0.0385	0.0188 ± 0.0141	0.0160 ± 0.0102
PCA-MML^{ANG}	0.0321 ± 0.0224	0.0166 ± 0.0043	0.0138 ± 0.0021
MML_{NU}^{WT}	0.0463 ± 0.0449	0.0242 ± 0.0143	0.0160 ± 0.0072
MML_{SR}^{WT}	0.0298 ± 0.0114	0.0166 ± 0.0043	0.0138 ± 0.0021
SVM^{light}	0.0392 ± 0.0262	0.0182 ± 0.0070	0.0143 ± 0.0031
Lagrangian	0.0754 ± 0.0709	0.0231 ± 0.0186	0.0136 ± 0.0083
SMOBR	0.0322 ± 0.0160	0.0199 ± 0.0116	0.0148 ± 0.0053
85/55			
PCA-MML$_{NU}^{WT}$	0.0565 ± 0.0204	0.0507 ± 0.0111	0.0449 ± 0.0092
PCA-MML$_{SR}^{WT}$	0.0597 ± 0.0289	0.0518 ± 0.0127	0.0442 ± 0.0079
PCA-MML$_{NUR}^{WT}$	0.0565 ± 0.0204	0.0512 ± 0.0115	0.0447 ± 0.0103
PCA-MML^{ANG}	0.0572 ± 0.0225	0.0509 ± 0.0189	0.0435 ± 0.0058
MML_{NU}^{WT}	0.0608 ± 0.0267	0.0453 ± 0.0126	0.0360 ± 0.0088
MML_{SR}^{WT}	0.0565 ± 0.0204	0.0490 ± 0.0085	0.0427 ± 0.0045
SVM^{light}	0.0563 ± 0.0203	0.0490 ± 0.0087	0.0424 ± 0.0046
Lagrangian	0.0816 ± 0.0675	0.0488 ± 0.0130	0.0424 ± 0.0048
SMOBR	0.0616 ± 0.0280	0.0473 ± 0.0113	0.0393 ± 0.0066
95/05			
PCA-MML$_{NU}^{WT}$	0.1544 ± 0.0359	0.1032 ± 0.0565	0.0670 ± 0.0372
PCA-MML$_{SR}^{WT}$	0.1509 ± 0.0335	0.0775 ± 0.0462	0.0451 ± 0.0336
PCA-MML$_{NUR}^{WT}$	0.1607 ± 0.0449	0.0885 ± 0.0559	0.0575 ± 0.0330
PCA-MML^{ANG}	0.1366 ± 0.0443	0.0629 ± 0.0516	0.0540 ± 0.0362
MML_{NU}^{WT}	0.1473 ± 0.0274	0.1715 ± 0.0601	0.0565 ± 0.0464
MML_{SR}^{WT}	0.1579 ± 0.0333	0.2841 ± 0.0239	0.3112 ± 0.0083
SVM^{light}	0.1444 ± 0.0220	0.2435 ± 0.0306	0.2705 ± 0.0483
Lagrangian	0.1226 ± 0.0713	0.0924 ± 0.0533	0.0670 ± 0.0308
SMOBR	0.1575 ± 0.0332	0.2834 ± 0.0251	0.3112 ± 0.0083
50/50			
PCA-MML$_{NU}^{WT}$	0.0140 ± 0.0531	0.0035 ± 0.0155	0.0013 ± 0.0064
PCA-MML$_{SR}^{WT}$	0.0139 ± 0.0370	0.0020 ± 0.0158	0.0002 ± 0.0002
PCA-MML$_{NUR}^{WT}$	0.0162 ± 0.0429	0.0027 ± 0.0174	0.0013 ± 0.0067
PCA-MML^{ANG}	0.0086 ± 0.0218	0.0004 ± 0.0005	0.0002 ± 0.0002
MML_{NU}^{WT}	0.0214 ± 0.0329	0.0063 ± 0.0157	0.0042 ± 0.0071
MML_{SR}^{WT}	0.0065 ± 0.0065	0.0004 ± 0.0005	0.0002 ± 0.0002
SVM^{light}	0.0417 ± 0.0492	0.0076 ± 0.0104	0.0028 ± 0.0048
Lagrangian	0.0636 ± 0.0769	0.0136 ± 0.0183	0.0060 ± 0.0071
SMOBR	0.0079 ± 0.0075	0.0029 ± 0.0056	0.0032 ± 0.0078

Table 5. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0125 ± 0.0023	0.0068 ± 0.0034
PCA-MML$_{SR}^{WT}$	0.0131 ± 0.0021	0.0088 ± 0.0040
PCA-MML$_{NUR}^{WT}$	0.0123 ± 0.0028	0.0068 ± 0.0037
PCA-MML^{ANG}	0.0128 ± 0.0019	0.0068 ± 0.0048
MML_{NU}^{WT}	0.0106 ± 0.0018	0.0092 ± 0.0012
MML_{SR}^{WT}	0.0126 ± 0.0004	0.0125 ± 0.0002
SVM^{light}	0.0126 ± 0.0004	0.0125 ± 0.0002
Lagrangian	0.0071 ± 0.0033	0.0054 ± 0.0026
SMOBR	0.0114 ± 0.0018	0.0110 ± 0.0015
85/55		
PCA-MML$_{NU}^{WT}$	0.0122 ± 0.0052	0.0067 ± 0.0042
PCA-MML$_{SR}^{WT}$	0.0104 ± 0.0059	0.0050 ± 0.0036
PCA-MML$_{NUR}^{WT}$	0.0107 ± 0.0043	0.0074 ± 0.0039
PCA-MML^{ANG}	0.0069 ± 0.0060	0.0032 ± 0.0026
MML_{NU}^{WT}	0.0207 ± 0.0090	0.0103 ± 0.0095
MML_{SR}^{WT}	0.0379 ± 0.0008	0.0372 ± 0.0004
SVM^{light}	0.0379 ± 0.0008	0.0372 ± 0.0004
Lagrangian	0.0379 ± 0.0008	0.0372 ± 0.0004
SMOBR	0.0340 ± 0.0041	0.0340 ± 0.0033
95/05		
PCA-MML$_{NU}^{WT}$	0.0291 ± 0.0203	0.0203 ± 0.0139
PCA-MML$_{SR}^{WT}$	0.0113 ± 0.0093	0.0042 ± 0.0043
PCA-MML$_{NUR}^{WT}$	0.0303 ± 0.0183	0.0252 ± 0.0116
PCA-MML^{ANG}	0.0179 ± 0.0163	0.0106 ± 0.0087
MML_{NU}^{WT}	0.0293 ± 0.0147	0.0272 ± 0.0124
MML_{SR}^{WT}	0.3236 ± 0.0017	0.3255 ± 0.0011
SVM^{light}	0.2606 ± 0.0505	0.2692 ± 0.0543
Lagrangian	0.0528 ± 0.0222	0.0504 ± 0.0172
SMOBR	0.3236 ± 0.0017	0.3255 ± 0.0011
50/50		
PCA-MML$_{NU}^{WT}$	0.0001 ± 0.0011	1.5790e-6 ± 1.6290e-6
PCA-MML$_{SR}^{WT}$	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
PCA-MML$_{NUR}^{WT}$	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
PCA-MML^{ANG}	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
MML_{NU}^{WT}	0.0012 ± 0.0016	0.0006 ± 0.0007
MML_{SR}^{WT}	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
SVM^{light}	0.0008 ± 0.0010	0.0004 ± 0.0005
Lagrangian	0.0014 ± 0.0014	0.0006 ± 0.0007
SMOBR	0.0008 ± 0.0015	0.0003 ± 0.0006

Table 6. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

B Kullback-Leibler Tables for Artificial Binomial Data, Translation Matrix 2

The following tables show the Kullback-Leibler distances between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, two of the Spikey methods ([16, 17]) and the SVMs, SVM^{light}, the Lagrangian SVM and SMOBR for artificial Binary data skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0467 ± 0.0405	0.0224 ± 0.0124	0.0203 ± 0.0115
PCA-MML$_{SR}^{WT}$	0.0417 ± 0.0399	0.0205 ± 0.0095	0.0191 ± 0.0098
PCA-MML$_{NUR}^{WT}$	0.0457 ± 0.0423	0.0226 ± 0.0125	0.0196 ± 0.0088
PCA-MML^{ANG}	0.0405 ± 0.0378	0.0196 ± 0.0033	0.0182 ± 0.0101
MML_{NU}^{WT}	0.0500 ± 0.0401	0.0234 ± 0.0129	0.0191 ± 0.0123
MML_{SR}^{WT}	0.0346 ± 0.0128	0.0196 ± 0.0033	0.0172 ± 0.0017
SVM^{light}	0.0465 ±0.0302 ^{*2}	0.0200 ±0.0066 ^{*7}	0.0166 ±0.0035 ^{*8}
Lagrangian	0.0654 ± 0.0568	0.0211 ± 0.0133	0.0159 ± 0.0092
SMOBR	0.0339 ± 0.0188	0.0184 ± 0.0074	0.0159 ± 0.0049
85/55			
PCA-MML$_{NU}^{WT}$	0.0576 ± 0.0653	0.0460 ± 0.0173	0.0411 ± 0.0109
PCA-MML$_{SR}^{WT}$	0.0540 ± 0.0634	0.0450 ± 0.0200	0.0420 ± 0.0141
PCA-MML$_{NUR}^{WT}$	0.0565 ± 0.0648	0.0480 ± 0.0238	0.0413 ± 0.0102
PCA-MML^{ANG}	0.0522 ± 0.0623	0.0481 ± 0.0305	0.0419 ± 0.0150
MML_{NU}^{WT}	0.0577 ± 0.0554	0.0433 ± 0.0218	0.0334 ± 0.0121
MML_{SR}^{WT}	0.0400 ± 0.0286	0.0410 ± 0.0071	0.0389 ± 0.0042
SVM^{light}	0.0546 ±0.0424 ^{*5}	0.0439 ±0.0142 ^{*1}	0.0395 ±0.0058 ^{*1}
Lagrangian	0.0926 ± 0.0835	0.0482 ± 0.0174	0.0422 ± 0.0089
SMOBR	0.0393 ± 0.0305	0.0390 ± 0.0141	0.0349 ± 0.0079
95/05			
PCA-MML$_{NU}^{WT}$	0.2081 ± 0.0527	0.1092 ± 0.0534	0.0844 ± 0.0462
PCA-MML$_{SR}^{WT}$	0.2038 ± 0.0535	0.0882 ± 0.0536	0.0547 ± 0.0469
PCA-MML$_{NUR}^{WT}$	0.2110 ± 0.0464	0.1082 ± 0.0558	0.0837 ± 0.0441
PCA-MML^{ANG}	0.1801 ± 0.0735	0.0661 ± 0.0479	0.0767 ± 0.0502
MML_{NU}^{WT}	0.1805 ± 0.0450	0.1285 ± 0.0422	0.1139 ± 0.0258
MML_{SR}^{WT}	0.2167 ± 0.0409	0.3652 ± 0.0208	0.3679 ± 0.0109
SVM^{light}	0.1884 ± 0.0280	0.2933 ±0.0472 ^{*2}	0.2609 ± 0.0432
Lagrangian	0.0573 ± 0.0434	0.0932 ± 0.0490	0.0690 ± 0.0349
SMOBR	0.1411 ± 0.0673	0.1174 ± 0.0535	0.1093 ± 0.0501
50/50			
PCA-MML$_{NU}^{WT}$	0.0108 ± 0.0356	0.0048 ± 0.0247	0.0001 ± 0.0001
PCA-MML$_{SR}^{WT}$	0.0072 ± 0.0068	0.0032 ± 0.0201	0.0001 ± 0.0001
PCA-MML$_{NUR}^{WT}$	0.0166 ± 0.0483	0.0052 ± 0.0248	0.0001 ± 0.0001
PCA-MML^{ANG}	0.0108 ± 0.0356	0.0020 ± 0.0161	0.0001 ± 0.0001
MML_{NU}^{WT}	0.0252 ± 0.0380	0.0122 ± 0.0206	0.0053 ± 0.0089
MML_{SR}^{WT}	0.0072 ± 0.0068	0.0004 ± 0.0005	0.0001 ± 0.0001
SVM^{light}	0.0273 ±0.0291 ^{*1}	0.0076 ± 0.0088	0.0025 ±0.0035 ^{*2}
Lagrangian	0.0569 ± 0.0667	0.0121 ± 0.0177	0.0067 ± 0.0083
SMOBR	0.0115 ± 0.0167	0.0022 ± 0.0057	0.0011 ± 0.0027

Table 7. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 2 - data sets sizes 10 to 100 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0127 ± 0.0051	0.0043 ± 0.0034
PCA-MML$_{SR}^{WT}$	0.0155 ± 0.0036	0.0069 ± 0.0048
PCA-MML$_{NUR}^{WT}$	0.0146 ± 0.0033	0.0072 ± 0.0034
PCA-MML^{ANG}	0.0158 ± 0.0021	0.0036 ± 0.0036
MML_{NU}^{WT}	0.0096 ± 0.0035	0.0058 ± 0.0031
MML_{SR}^{WT}	0.0156 ± 0.000	0.0154 ± 0.0002
SVM^{light}	0.0144 ±0.0025 ^{*16}	0.0148 ±0.0018 ^{*18}
Lagrangian	0.0068 ± 0.0034	0.0047 ± 0.0027
SMOBR	0.0126 ± 0.0041	0.0122 ± 0.0041
85/55		
PCA-MML$_{NU}^{WT}$	0.0097 ± 0.0076	0.0049 ± 0.0043
PCA-MML$_{SR}^{WT}$	0.0099 ± 0.0076	0.0038 ± 0.0045
PCA-MML$_{NUR}^{WT}$	0.0151 ± 0.0069	0.0095 ± 0.0037
PCA-MML^{ANG}	0.0082 ± 0.0071	0.0032 ± 0.0028
MML_{NU}^{WT}	0.0130 ± 0.0062	0.0107 ± 0.0019
MML_{SR}^{WT}	0.0400 ± 0.0008	0.0401 ± 0.0004
SVM^{light}	0.0400 ±0.0008 ^{*4}	0.0401 ±0.0004 ^{*7}
Lagrangian	0.0391 ± 0.0011	0.0390 ± 0.0008
SMOBR	0.0377 ± 0.0030	0.0387 ± 0.0040
95/05		
PCA-MML$_{NU}^{WT}$	0.0343 ± 0.0250	0.0303 ± 0.0264
PCA-MML$_{SR}^{WT}$	0.0116 ± 0.0143	0.0056 ± 0.0050
PCA-MML$_{NUR}^{WT}$	0.0611 ± 0.0193	0.0679 ± 0.0124
PCA-MML^{ANG}	0.0220 ± 0.0148	0.0122 ± 0.0122
MML_{NU}^{WT}	0.1064 ± 0.0055	0.1055 ± 0.0029
MML_{SR}^{WT}	0.3883 ± 0.0019	0.3902 ± 0.0010
SVM^{light}	0.2012 ±0.0093 ^{*1}	0.1597 ±0.0136 ^{*2}
Lagrangian	0.0389 ± 0.0190	0.0296 ± 0.0156
SMOBR	0.1257 ± 0.0404	0.1282 ± 0.0236
50/50		
PCA-MML$_{NU}^{WT}$	0.0044 ± 0.0129	0.0001 ± 0.0006
PCA-MML$_{SR}^{WT}$	0.0041 ± 0.0124	1.3644e-6 ± 1.3243e-6
PCA-MML$_{NUR}^{WT}$	0.0040 ± 0.0122	0.0001 ± 0.0006
PCA-MML^{ANG}	0.0041 ± 0.0124	1.3644e-6 ± 1.3243e-6
MML_{NU}^{WT}	0.0046 ± 0.0103	0.0006 ± 0.0008
MML_{SR}^{WT}	0.0041 ± 0.0124	1.3644 ± 1.3243
SVM^{light}	0.0050 ±0.0122 ^{*11}	0.0005 ±0.0006 ^{*11}
Lagrangian	0.0060 ± 0.0140	0.0006 ± 0.0008
SMOBR	0.0038 ± 0.0110	0.0002 ± 0.0004

Table 8. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 2 - data sets sizes 500 to 1000 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0542 ± 0.0697	0.0196 ± 0.0141	0.0160 ± 0.0077
PCA-MML$_{SR}^{WT}$	0.0350 ± 0.0335	0.0175 ± 0.0099	0.0157 ± 0.0099
PCA-MML$_{NUR}^{WT}$	0.0461 ± 0.0573	0.0190 ± 0.0125	0.0164 ± 0.0099
PCA-MML^{ANG}	0.0382 ± 0.0450	0.0165 ± 0.0037	0.0161 ± 0.0139
MML_{NU}^{WT}	0.0457 ± 0.0463	0.0265 ± 0.0177	0.0180 ± 0.0094
MML_{SR}^{WT}	0.0288 ± 0.0121	0.0165 ± 0.0037	0.0148 ± 0.0019
SVM^{light}	0.0480 ±0.0421 ^{*1}	0.0194 ±0.0103 ^{*2}	0.0154 ±0.0035 ^{*3}
Lagrangian	0.0791 ± 0.0849	0.0210 ± 0.0158	0.0152 ± 0.0123
SMOBR	0.0341 ± 0.0174	0.0174 ± 0.0049	0.0157 ± 0.0041
85/55			
PCA-MML$_{NU}^{WT}$	0.0593 ± 0.0725	0.0412 ± 0.0270	0.0341 ± 0.0084
PCA-MML$_{SR}^{WT}$	0.0536 ± 0.0673	0.0357 ± 0.0207	0.0337 ± 0.0095
PCA-MML$_{NUR}^{WT}$	0.0602 ± 0.0694	0.0396 ± 0.0216	0.0335 ± 0.0087
PCA-MML^{ANG}	0.0568 ± 0.0808	0.0335 ± 0.0133	0.0325 ± 0.0066
MML_{NU}^{WT}	0.0589 ± 0.0687	0.0410 ± 0.0232	0.0339 ± 0.0096
MML_{SR}^{WT}	0.0335 ± 0.0290	0.0325 ± 0.0067	0.0321 ± 0.0041
SVM^{light}	0.0537 ± 0.0463	0.0345 ±0.0138 ^{*2}	0.0322 ±0.0044 ^{*5}
Lagrangian	0.0935 ± 0.0922	0.0228 ± 0.0209	0.0134 ± 0.0118
SMOBR	0.0427 ± 0.0368	0.0355 ± 0.0122	0.0338 ± 0.0067
95/05			
PCA-MML$_{NU}^{WT}$	0.1716 ± 0.0504	0.0859 ± 0.0685	0.0655 ± 0.0444
PCA-MML$_{SR}^{WT}$	0.1735 ± 0.0581	0.0568 ± 0.0616	0.0342 ± 0.0258
PCA-MML$_{NUR}^{WT}$	0.1832 ± 0.0649	0.0807 ± 0.0569	0.0564 ± 0.0352
PCA-MML^{ANG}	0.1477 ± 0.0566	0.0550 ± 0.0470	0.0713 ± 0.0522
MML_{NU}^{WT}	0.1803 ± 0.0501	0.2400 ± 0.0285	0.2198 ± 0.0390
MML_{SR}^{WT}	0.1770 ± 0.0447	0.3046 ± 0.0264	0.3220 ± 0.0099
SVM^{light}	0.1750 ± 0.0471	0.2911 ±0.0233 ^{*2}	0.3112 ±0.0265 ^{*2}
Lagrangian	0.2097 ± 0.1029	0.3281 ± 0.1307	0.3595 ± 0.0769
SMOBR	0.1779 ± 0.0475	0.3045 ± 0.0265	0.3221 ± 0.0099
50/50			
PCA-MML$_{NU}^{WT}$	0.0122 ± 0.0498	0.0057 ± 0.0239	0.0006 ± 0.0050
PCA-MML$_{SR}^{WT}$	0.0050 ± 0.0060	0.0004 ± 0.0005	0.0006 ± 0.0049
PCA-MML$_{NUR}^{WT}$	0.0089 ± 0.0283	0.0049 ± 0.0202	0.0001 ± 0.0002
PCA-MML^{ANG}	0.0086 ± 0.0357	0.0021 ± 0.0165	0.0001 ± 0.0002
MML_{NU}^{WT}	0.0249 ± 0.0495	0.0109 ± 0.0237	0.0073 ± 0.0106
MML_{SR}^{WT}	0.0050 ± 0.0060	0.0004 ± 0.0005	0.0001 ± 0.0002
SVM^{light}	0.0329 ±0.0344 ^{*1}	0.0077 ±0.0128 ^{*1}	0.0047 ±0.0052 ^{*2}
Lagrangian	0.0522 ± 0.0549	0.0141 ± 0.0174	0.0073 ± 0.0070
SMOBR	0.0186 ± 0.0316	0.0056 ± 0.0109	0.0030 ± 0.0058

Table 9. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 2 - data set sizes 10 to 100. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0124 ± 0.0031	0.0044 ± 0.0035
PCA-MML$_{SR}^{WT}$	0.0137 ± 0.0038	0.0028 ± 0.0031
PCA-MML$_{NUR}^{WT}$	0.0125 ± 0.0033	0.0047 ± 0.0038
PCA-MML^{ANG}	0.0134 ± 0.0015	0.0069 ± 0.0046
MML_{NU}^{WT}	0.0119 ± 0.0018	0.0112 ± 0.0018
MML_{SR}^{WT}	0.0133 ± 0.0004	0.0131 ± 0.0002
SVM^{light}	0.0133 ± 0.0004	0.0131 ±0.0002 ^{*14}
Lagrangian	0.0091 ± 0.0042	0.0091 ± 0.0033
SMOBR	0.0132 ± 0.0007	0.0126 ± 0.0010
85/55		
PCA-MML$_{NU}^{WT}$	0.0089 ± 0.0072	0.0049 ± 0.0045
PCA-MML$_{SR}^{WT}$	0.0061 ± 0.0075	0.0026 ± 0.0027
PCA-MML$_{NUR}^{WT}$	0.0090 ± 0.0076	0.0037 ± 0.0038
PCA-MML^{ANG}	0.0088 ± 0.0073	0.0038 ± 0.0028
MML_{NU}^{WT}	0.0288 ± 0.0028	0.0230 ± 0.0058
MML_{SR}^{WT}	0.0335 ± 0.0007	0.0335 ± 0.0004
SVM^{light}	0.0335 ±0.0007 ^{*5}	0.0335 ±0.0004 ^{*2}
Lagrangian	0.0046 ± 0.0041	0.0037 ± 0.0025
SMOBR	0.0339 ± 0.0016	0.0337 ± 0.0006
95/05		
PCA-MML$_{NU}^{WT}$	0.0372 ± 0.0237	0.0352 ± 0.0206
PCA-MML$_{SR}^{WT}$	0.0119 ± 0.0093	0.0061 ± 0.0043
PCA-MML$_{NUR}^{WT}$	0.0300 ± 0.0207	0.0231 ± 0.0149
PCA-MML^{ANG}	0.0240 ± 0.0182	0.0147 ± 0.0149
MML_{NU}^{WT}	0.1686 ± 0.0112	0.1643 ± 0.0059
MML_{SR}^{WT}	0.3366 ± 0.0022	0.3388 ± 0.0010
SVM^{light}	0.2858 ±0.0596 ^{*8}	0.1742 ±0.0430 ^{*13}
Lagrangian	0.4127 ± 0.0419	0.4279 ± 0.0312
SMOBR	0.3366 ± 0.0022	0.3388 ± 0.0010
50/50		
PCA-MML$_{NU}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML$_{SR}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML$_{NUR}^{WT}$	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
PCA-MML^{ANG}	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
MML_{NU}^{WT}	0.0014 ± 0.0020	0.0008 ± 0.0011
MML_{SR}^{WT}	6.2531e-6 ± 6.3740e-6	1.4355e-6 ± 1.5821e-6
SVM^{light}	0.0007 ±0.0009 ^{*5}	0.0003 ±0.0004 ^{*7}
Lagrangian	0.0014 ± 0.0012	0.0007 ± 0.0006
SMOBR	0.0006 ± 0.0013	0.0004 ± 0.0007

Table 10. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 2 - data set sizes 500 to 1000. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.0487 ± 0.0590	0.0187 ± 0.0103	0.0157 ± 0.0112
PCA-MML$_{SR}^{WT}$	0.0313 ± 0.0169	0.0181 ± 0.0153	0.0152 ± 0.0107
PCA-MML$_{NUR}^{WT}$	0.0388 ± 0.0390	0.0189 ± 0.0147	0.0150 ± 0.0061
PCA-MML^{ANG}	0.0321 ± 0.0224	0.0166 ± 0.0043	0.0138 ± 0.0021
MML_{NU}^{WT}	0.0496 ± 0.0492	0.0268 ± 0.0179	0.0166 ± 0.0086
MML_{SR}^{WT}	0.0298 ± 0.0114	0.0166 ± 0.0043	0.0138 ± 0.0021
SVM^{light}	0.0403 ± 0.0288	0.0187 ± 0.0085	0.0138 ±0.0021 ^{*2}
Lagrangian	0.0747 ± 0.0698	0.0204 ± 0.0217	0.0132 ± 0.0119
SMOBR	0.0395 ± 0.0313	0.0203 ± 0.0114	0.0143 ± 0.0035
85/55			
PCA-MML$_{NU}^{WT}$	0.0565 ± 0.0204	0.0506 ± 0.0116	0.0443 ± 0.0084
PCA-MML$_{SR}^{WT}$	0.0599 ± 0.0299	0.0517 ± 0.0125	0.0445 ± 0.0073
PCA-MML$_{NUR}^{WT}$	0.0584 ± 0.0273	0.0507 ± 0.0118	0.0449 ± 0.0114
PCA-MML^{ANG}	0.0572 ± 0.0225	0.0509 ± 0.0189	0.0435 ± 0.0058
MML_{NU}^{WT}	0.0602 ± 0.0252	0.0450 ± 0.0142	0.0351 ± 0.0103
MML_{SR}^{WT}	0.0565 ± 0.0204	0.0490 ± 0.0085	0.0427 ± 0.0045
SVM^{light}	0.0566 ±0.0201 ^{*2}	0.0490 ±0.0085 ^{*1}	0.0426 ±0.0045 ^{*1}
Lagrangian	0.0785 ± 0.0716	0.0179 ± 0.0210	0.0067 ± 0.0090
SMOBR	0.0607 ± 0.0275	0.0482 ± 0.0132	0.0386 ± 0.0069
95/05			
PCA-MML$_{NU}^{WT}$	0.1522 ± 0.0360	0.0941 ± 0.0576	0.0602 ± 0.0323
PCA-MML$_{SR}^{WT}$	0.1498 ± 0.0309	0.0690 ± 0.0474	0.0467 ± 0.0292
PCA-MML$_{NUR}^{WT}$	0.1590 ± 0.0426	0.0877 ± 0.0578	0.0563 ± 0.0280
PCA-MML^{ANG}	0.1346 ± 0.0432	0.0627 ± 0.0517	0.0546 ± 0.0370
MML_{NU}^{WT}	0.1436 ± 0.0293	0.1662 ± 0.0451	0.0915 ± 0.0447
MML_{SR}^{WT}	0.1579 ± 0.0333	0.2841 ± 0.0239	0.3112 ± 0.0083
SVM^{light}	0.1429 ± 0.0223	0.2344 ±0.0387 ^{*1}	0.2503 ±0.0592 ^{*1}
Lagrangian	0.1627 ± 0.0915	0.2967 ± 0.1321	0.3486 ± 0.0696
SMOBR	0.1579 ± 0.0333	0.2841 ± 0.0239	0.3112 ± 0.0083
50/50			
PCA-MML$_{NU}^{WT}$	0.0173 ± 0.0618	0.0035 ± 0.0155	0.0013 ± 0.0064
PCA-MML$_{SR}^{WT}$	0.0157 ± 0.0407	0.0020 ± 0.0158	0.0002 ± 0.0002
PCA-MML$_{NUR}^{WT}$	0.0196 ± 0.0487	0.0028 ± 0.0140	0.0006 ± 0.0044
PCA-MML^{ANG}	0.0086 ± 0.0218	0.0004 ± 0.0005	0.0002 ± 0.0002
MML_{NU}^{WT}	0.0245 ± 0.0354	0.0082 ± 0.0191	0.0061 ± 0.0107
MML_{SR}^{WT}	0.0065 ± 0.0065	0.0004 ± 0.0005	0.0002 ± 0.0002
SVM^{light}	0.0401 ± 0.0452	0.0086 ± 0.0114	0.0032 ± 0.0048
Lagrangian	0.0599 ± 0.0633	0.0157 ± 0.0159	0.0072 ± 0.0064
SMOBR	0.0148 ± 0.0230	0.0045 ± 0.0092	0.0026 ± 0.0041

Table 11. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Translation Matrix 2 - data set sizes 10 to 100. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.0124 ± 0.0025	0.0069 ± 0.0033
PCA-MML$_{SR}^{WT}$	0.0131 ± 0.0021	0.0089 ± 0.0040
PCA-MML$_{NUR}^{WT}$	0.0125 ± 0.0024	0.0063 ± 0.0036
PCA-MML^{ANG}	0.0128 ± 0.0019	0.0068 ± 0.0048
MML_{NU}^{WT}	0.0101 ± 0.0024	0.0087 ± 0.0018
MML_{SR}^{WT}	0.0126 ± 0.0004	0.0125 ± 0.0002
SVM^{light}	0.0126 ±0.0004 ^{*4}	0.0125 ±0.0002 ^{*9}
Lagrangian	0.0078 ± 0.0042	0.0085 ± 0.0035
SMOBR	0.0117 ± 0.0016	0.0113 ± 0.0016
85/55		
PCA-MML$_{NU}^{WT}$	0.0108 ± 0.0065	0.0058 ± 0.0044
PCA-MML$_{SR}^{WT}$	0.0095 ± 0.0068	0.0043 ± 0.0032
PCA-MML$_{NUR}^{WT}$	0.0108 ± 0.0057	0.0071 ± 0.0032
PCA-MML^{ANG}	0.0069 ± 0.0059	0.0031 ± 0.0026
MML_{NU}^{WT}	0.0205 ± 0.0066	0.0103 ± 0.0055
MML_{SR}^{WT}	0.0379 ± 0.0008	0.0372 ± 0.0004
SVM^{light}	0.0379 ±0.0008 ^{*3}	0.0372 ±0.0004 ^{*3}
Lagrangian	0.0024 ± 0.0026	0.0016 ± 0.0016
SMOBR	0.0321 ± 0.0047	0.0315 ± 0.0044
95/05		
PCA-MML$_{NU}^{WT}$	0.0280 ± 0.0213	0.0200 ± 0.0138
PCA-MML$_{SR}^{WT}$	0.0081 ± 0.0068	0.0045 ± 0.0031
PCA-MML$_{NUR}^{WT}$	0.0291 ± 0.0185	0.0258 ± 0.0135
PCA-MML^{ANG}	0.0179 ± 0.0163	0.0106 ± 0.0087
MML_{NU}^{WT}	0.0488 ± 0.0092	0.0435 ± 0.0046
MML_{SR}^{WT}	0.3236 ± 0.0017	0.3255 ± 0.0011
SVM^{light}	0.1757 ±0.0455 ^{*11}	0.1160 ±0.0139 ^{*20}
Lagrangian	0.3688 ± 0.0458	0.3722 ± 0.0315
SMOBR	0.3236 ± 0.0017	0.3255 ± 0.0011
50/50		
PCA-MML$_{NU}^{WT}$	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
PCA-MML$_{SR}^{WT}$	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
PCA-MML$_{NUR}^{WT}$	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
PCA-MML^{ANG}	7.0925e-6 ± 6.6002e-6	1.5790e-6 ± 1.6290e-6
MML_{NU}^{WT}	0.0012 ± 0.0016	0.0007 ± 0.0008
MML_{SR}^{WT}	7.0925e-6 ± 6.6002e-6	1.5791e-6 ± 1.6290e-6
SVM^{light}	0.0008 ±0.0010 ^{*3}	0.0004 ±0.0005 ^{*3}
Lagrangian	0.0016 ± 0.0013	0.0007 ± 0.0006
SMOBR	0.0007 ± 0.0011	0.0003 ± 0.0005

Table 12. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Translation Matrix 2 - data set sizes 500 to 1000. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

PCA-Spikey Results Tables for Binomial - Area Between True and Inferred Hyperplanes

C PCA-Spikey Area Tables for Artificial Binomial Data, Translation Matrix 1

The following tables show the Area between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, two of the Spikey methods ([16, 17]) and the SVMs, SVM^{light}, the Lagrangian SVM and SMOBR for artificial Binary data skewed via Translation Matrix 1.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3019 ± 0.0843, 91	0.4106 ± 0.0756, 94	0.3277 ± 0.0919, 90
PCA-MML$_{SR}^{WT}$	0.2905 ± 0.0765, 96	0.3016 ± 0.1964, 98	0.3992 ± 0.0838, 97
PCA-MML$_{NUR}^{WT}$	0.2905 ± 0.0577, 90	0.3675 ± 0.0813, 94	0.3082 ± 0.0679, 91
PCA-MML^{ANG}	0.2869 ± 0.0215,97	N/A ± N/A, 100	0.4913 ± N/A, 99
MML_{NU}^{WT}	0.2877 ± 0.1004, 81	0.2681 ± 0.0882, 55	0.2620 ± 0.0742, 27
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2358 ±0.0500 ^{*1}, 0	0.2334 ± 0.0298, 0	0.2304 ±0.0254 ^{*35}, 0
Lagrangian	0.2669 ± 0.1102, 0	0.2077 ± 0.1020, 0	0.1878 ± 0.0922, 0
SMOBR	0.2272 ± 0.0518, 0	0.2356 ± 0.0891, 0	0.2230 ± 0.0866, 0
85/55			
PCA-MML$_{NU}^{WT}$	0.2953 ± 0.0844, 86	0.2847 ± 0.0987, 83	0.2482 ± 0.0930, 70
PCA-MML$_{SR}^{WT}$	0.3769 ± 0.0979, 93	0.2806 ± 0.0720, 94	0.2399 ± 0.1098, 88
PCA-MML$_{NUR}^{WT}$	0.3227 ± 0.0953, 91	0.2576 ± 0.0639, 84	0.2677 ± 0.1073, 74
PCA-MML^{ANG}	0.3730 ± 0.0617, 95	0.3125 ± 0.1292, 93	0.2878 ± 0.1274, 95
MML_{NU}^{WT}	0.2570 ± 0.0837, 76	0.2523 ± 0.0677, 37	0.2630 ± 0.0815, 8
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2994 ±0.0801 ^{*4}, 0	0.2654 ±0.0449 ^{*9}, 0	0.2608 ±0.0317 ^{*14}, 0
Lagrangian	0.3682 ± 0.0900, 0	0.4066 ± 0.0769, 0	0.400 ± 0.0806, 0
SMOBR	0.2435 ± 0.0581, 0	0.2371 ± 0.0803, 0	0.2434 ± 0.0886, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.2030 ± 0.0880, 70	0.0617 ± 0.0387, 0	0.0403 ± 0.0247, 0
PCA-MML$_{SR}^{WT}$	0.1548 ± 0.0591, 74	0.0432 ± 0.0263, 0	0.0214 ± 0.0209, 0
PCA-MML$_{NUR}^{WT}$	0.2010 ± 0.0605, 77	0.0570 ± 0.0397, 0	0.0382 ± 0.0213, 0
PCA-MML^{ANG}	0.1076 ± 0.0881, 64	0.0301 ± 0.0246, 0	0.0360 ± 0.0281, 0
MML_{NU}^{WT}	0.2035 ± 0.0630, 48	0.1132 ± 0.0863, 0	0.0491 ± 0.0525, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2205 ±0.0428 ^{*1}, 0	0.1861 ±0.0420 ^{*6}, 0	0.1606 ±0.0233 ^{*3}, 0
Lagrangian	0.0522 ± 0.0498, 0	0.0467 ± 0.0269, 0	0.0329 ± 0.0179, 0
SMOBR	0.2208 ± 0.0655, 3	0.1443 ± 0.0484, 0	0.1040 ± 0.0369, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.2670 ± N/A, 99	0.3705 ± 0.0990, 98	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	0.4399 ± 0.0432, 97	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.3031 ± 0.1123, 97	0.4697 ± 0.0247, 97	0.1783 ± N/A, 99
PCA-MML^{ANG}	N/A ± N/A, 100	0.4199 ± N/A, 99	N/A ± N/A, 100
MML_{NU}^{WT}	0.3252 ± 0.1171, 83	0.2964 ± 0.0958, 71	0.2673 ± 0.0501, 51
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2806 ±0.0849 ^{*1}, 0	0.2608 ±0.0281 ^{*7}, 0	0.2573 ±0.0215 ^{*6}, 0
Lagrangian	0.3429 ± 0.0968, 0	0.3521 ± 0.0863, 0	0.3348 ± 0.0771, 0
SMOBR	0.2405 ± 0.0285, 0	0.2291 ± 0.0455, 0	0.2251 ± 0.0515, 0

Table 13. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.1874 ± 0.1148, 38	0.0637 ± 0.0607, 1
PCA-MML$_{SR}^{WT}$	0.1785 ± 0.1160, 71	0.0207 ± 0.0684, 11
PCA-MML$_{NUR}^{WT}$	0.2132 ± 0.0887, 49	0.1120 ± 0.0738, 3
PCA-MML^{ANG}	0.2179 ± 0.1567, 95	0.0409 ± 0.0419, 1
MML_{NU}^{WT}	0.2291 ± 0.0644, 0	0.1837 ± 0.0787, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2337 ±0.0253 ^{*48}, 0	0.2445 ±0.0201 ^{*69}, 0
Lagrangian	0.0930 ± 0.0496, 0	0.0630 ± 0.0399, 0
SMOBR	0.2078 ± 0.0777, 0	0.5479 ± 0.3039, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0501 ± 0.0519, 0	0.0238 ± 0.0178, 0
PCA-MML$_{SR}^{WT}$	0.0509 ± 0.0530, 0	0.0207 ± 0.0175, 0
PCA-MML$_{NUR}^{WT}$	0.0751 ± 0.0516, 0	0.0497 ± 0.0288, 0
PCA-MML^{ANG}	0.0363 ± 0.0415, 0	0.0143 ± 0.0122, 0
MML_{NU}^{WT}	0.1227 ± 0.0965, 0	0.0510 ± 0.0684, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2531 ±2.7965 $*^{-}$ 16 ^{*33}, 0	0.2531 ±1.1231 $*^{-}$ 16 ^{*56}, 0
Lagrangian	0.3774 ± 0.0395, 0	0.3668 ± 0.0306, 0
SMOBR	0.2066 ± 0.0241, 0	0.2348 ± 0.0112, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0152 ± 0.0108, 0	0.0128 ± 0.0111, 0
PCA-MML$_{SR}^{WT}$	0.0040 ± 0.0035, 0	0.0022 ± 0.0019, 0
PCA-MML$_{NUR}^{WT}$	0.0282 ± 0.0111, 0	0.0315 ± 0.0085, 0
PCA-MML^{ANG}	0.0089 ± 0.0065, 0	0.0049 ± 0.0053, 0
MML_{NU}^{WT}	0.0203 ± 0.0239, 0	0.0165 ± 0.0100, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1496 ± 0.0106, 0	0.1433 ±0.0013 ^{*7}, 0
Lagrangian	0.0164 ± 0.0085, 0	0.0120 ± 0.0066, 0
SMOBR	0.0399 ± 0.0218, 0	0.0528 ± 0.0285, 0
50/50		
PCA-MML$_{NU}^{WT}$	0.2846 ± 0.0670, 97	0.2632 ± N/A, 99
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.2402 ± 0.0164, 98	0.2649 ± N/A, 99
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.2613 ± 0.0549, 2	0.2533 ± 0.0389, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2524 ±0.0148 ^{*14}, 0	0.2530 ±0.0072 ^{*24}, 0
Lagrangian	0.3538 ± 0.0823, 0	0.3526 ± 0.0817, 0
SMOBR	0.2297 ± 0.0686, 0	0.2452 ± 0.0494, 0

Table 14. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3435 ± 0.0927, 86	0.3250 ± 0.0730, 94	0.2922 ± 0.0625, 97
PCA-MML$_{SR}^{WT}$	0.3182 ± 0.1058, 96	0.4346 ± N/A, 99	0.4809 ± N/A, 99
PCA-MML$_{NUR}^{WT}$	0.2955 ± 0.0719, 90	0.2979 ± 0.0607, 95	0.3487 ± 0.0488, 96
PCA-MML^{ANG}	0.2893 ± 0.1029, 95	N/A ± N/A, 100	0.4100 ± N/A, 99
MML_{NU}^{WT}	0.3557 ± 0.0880, 88	0.3291 ± 0.0695, 51	0.3324 ± 0.0593, 31
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2676 ± 0.0718, 0	0.2271 ±0.0491 ^{*21}, 0	0.2215 ±0.0368 ^{*27}, 0
Lagrangian	0.2864 ± 0.1093, 0	0.2160 ± 0.1021, 0	0.1635 ± 0.0752, 0
SMOBR	0.2285 ± 0.0573, 0	0.2442 ± 0.0777, 0	0.2305 ± 0.0726, 0
85/55			
PCA-MML$_{NU}^{WT}$	0.3186 ± 0.0693, 85	0.3127 ± 0.0946, 83	0.3093 ± 0.0554, 83
PCA-MML$_{SR}^{WT}$	0.3300 ± 0.0935, 89	0.2749 ± 0.0808, 98	0.2871 ± 0.1228, 93
PCA-MML$_{NUR}^{WT}$	0.3209 ± 0.0696, 85	0.3108 ± 0.0989, 88	0.3027 ± 0.0898, 83
PCA-MML^{ANG}	0.3424 ± 0.0889, 90	0.3978 ± N/A, 99	0.2761 ± N/A, 99
MML_{NU}^{WT}	0.3173 ± 0.0768, 78	0.3224 ± 0.0758, 52	0.3201 ± 0.0578, 26
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2852 ± 0.0925, 0	0.2173 ±0.0623 ^{*7}, 0	0.2018 ±0.0155 ^{*22}, 0
Lagrangian	0.3727 ± 0.0828, 0	0.3843 ± 0.0903, 0	0.3578 ± 0.0873, 0
SMOBR	0.2249 ± 0.0577, 0	0.2600 ± 0.0943, 0	0.2818 ± 0.0859, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.2204 ± 0.1021, 79	0.0541 ± 0.0450, 1	0.0324 ± 0.0234, 0
PCA-MML$_{SR}^{WT}$	0.1738 ± 0.0864, 74	0.0260 ± 0.0253, 1	0.0156 ± 0.0115, 0
PCA-MML$_{NUR}^{WT}$	0.2539 ± 0.0998, 81	0.0381 ± 0.0279, 2	0.0290 ± 0.0201, 0
PCA-MML^{ANG}	0.1029 ± 0.0696, 63	0.0248 ± 0.0206, 1	0.0346 ± 0.0307, 0
MML_{NU}^{WT}	0.2867 ± 0.0811, 66	0.2421 ± 0.1102, 5	0.1066 ± 0.1359, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2143 ± 0.0347, 0	0.2003 ±0.0056 ^{*9}, 0	0.2004 ±0.0040 ^{*9}, 0
Lagrangian	0.1198 ± 0.0969, 0	0.0443 ± 0.0343, 0	0.0292 ± 0.0181, 0
SMOBR	0.2055 ± 0.0201, 13	0.2024 ± 0.0193, 0	0.2002 ± 0.0001, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.4018 ± 0.0588, 97	0.3830 ± 0.0856, 94	0.4687 ± N/A, 99
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	0.2018 ± N/A, 99	0.2748 ± N/A, 99
PCA-MML$_{NUR}^{WT}$	0.3193 ± 0.0233, 98	0.3595 ± 0.1148, 95	N/A ± N/A, 100
PCA-MML^{ANG}	0.4239 ± N/A, 99	0.4355 ± N/A, 99	N/A ± N/A, 100
MML_{NU}^{WT}	0.2795 ± 0.0380, 84	0.3122 ± 0.0764, 75	0.3188 ± 0.0444, 44
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.3333 ± 0.0636, 0	0.3259 ±0.0281 ^{*7}, 0	0.3264 ±0.0180 ^{*2}, 0
Lagrangian	0.3498 ± 0.0945, 0	0.3508 ± 0.0617, 0	0.3649 ± 0.0632, 0
SMOBR	0.2188 ± 0.0287, 0	0.2486 ± 0.0680, 0	0.2596 ± 0.0731, 0

Table 15. Area (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in $mean \pm standard\ deviation\ ^{*n}$ denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.1676 ± 0.1404, 67	0.0681 ± 0.0682, 2
PCA-MML$_{SR}^{WT}$	0.1516 ± 0.1120, 92	0.0404 ± 0.0454, 2
PCA-MML$_{NUR}^{WT}$	0.1606 ± 0.1091, 64	0.0886 ± 0.0901, 3
PCA-MML^{ANG}	0.1037 ± 0.0328, 96	0.0562 ± 0.0490, 27
MML_{NU}^{WT}	0.3111 ± 0.0595, 0	0.2966 ± 0.0784, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2000 ±8.4095 $*^{-}$ 17 ^{*49}, 0	0.2000 ±5.6656 $*^{-}$ 17 ^{*75}, 0
Lagrangian	0.0877 ± 0.0478, 0	0.0680 ± 0.0413, 0
SMOBR	0.2755 ± 0.0972, 3	0.2890 ± 0.0947, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0528 ± 0.0456, 0	0.0278 ± 0.0352, 0
PCA-MML$_{SR}^{WT}$	0.0296 ± 0.0310, 0	0.0172 ± 0.0127, 0
PCA-MML$_{NUR}^{WT}$	0.0467 ± 0.0482, 1	0.0217 ± 0.0227, 0
PCA-MML^{ANG}	0.0365 ± 0.0311, 1	0.0165 ± 0.0125, 0
MML_{NU}^{WT}	0.3082 ± 0.0964, 0	0.1534 ± 0.1411, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2000 ±2.7948 $*^{-}$ 16 ^{*27}, 0	0.2000 ±1.4008 $*^{-}$ 16 ^{*46}, 0
Lagrangian	0.3205 ± 0.0474, 0	0.3139 ± 0.0358, 0
SMOBR	0.2414 ± 0.0330, 0	0.2198 ± 0.0119, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0176 ± 0.0124, 0	0.0152 ± 0.0094, 0
PCA-MML$_{SR}^{WT}$	0.0049 ± 0.0039, 0	0.0025 ± 0.0026, 0
PCA-MML$_{NUR}^{WT}$	0.0140 ± 0.0097, 0	0.0108 ± 0.0076, 0
PCA-MML^{ANG}	0.0099 ± 0.0079, 0	0.0059 ± 0.0062, 0
MML_{NU}^{WT}	0.0177 ± 0.0119, 0	0.0142 ± 0.0077, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2000 ±3.0731 $*^{-}$ 16 ^{*23}, 0	0.1998 ±0.0015 ^{*27}, 0
Lagrangian	0.0144 ± 0.0083, 0	0.0123 ± 0.0065, 0
SMOBR	0.2001 ± 0.0000, 0	0.2001 ±6.9420 $*^{-}$ 6, 0
50/50		
PCA-MML$_{NU}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3263 ± 0.0438, 0	0.3191 ± 0.0371, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.3271 ±0.0275 ^{*21}, 0	0.3286 ±0.0198 ^{*31}, 0
Lagrangian	0.3644 ± 0.0688, 0	0.3693 ± 0.0681, 0
SMOBR	0.2762 ± 0.0989, 0	0.2546 ± 0.0822, 0

Table 16. Area (mean ± standard deviation) between the true hyperplane ($y = -0.9x + 130.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3697 ± 0.0899, 90	0.2520 ± 0.0843, 94	0.2846 ± 0.0464, 97
PCA-MML$_{SR}^{WT}$	0.1697 ± 0.0202, 98	0.2579 ± N/A, 99	0.2946 ± 0.0010, 98
PCA-MML$_{NUR}^{WT}$	0.3377 ± 0.0708, 94	0.3159 ± 0.1757, 97	0.2922 ± 0.0332, 94
PCA-MML^{ANG}	0.2360 ± 0.1522, 98	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3149 ± 0.1024, 80	0.2891 ± 0.0943, 51	0.3094 ± 0.0625, 30
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.2292 ±0.0697 ^{*2}, 0	0.1955 ±0.0243 ^{*8}, 0	0.1931 ±0.0177 ^{*17}, 0
Lagrangian	0.2975 ± 0.1073, 0	0.2253 ± 0.0982, 0	0.1710 ± 0.0912, 0
SMOBR	0.1994 ± 0.0503, 0	0.1996 ± 0.0502, 0	0.2017 ± 0.0584, 0
85/55			
PCA-MML$_{NU}^{WT}$	N/A ± N/A, 100	0.1998 ± 0.0856, 91	0.2889 ± 0.1129, 82
PCA-MML$_{SR}^{WT}$	0.3271 ± 0.0723, 98	0.1619 ± 0.0294, 91	0.2089 ± 0.1446, 90
PCA-MML$_{NUR}^{WT}$	N/A ± N/A, 100	0.2267 ± 0.0900, 90	0.2540 ± 0.1283, 82
PCA-MML^{ANG}	0.1707 ± N/A, 99	0.1843 ± 0.1221, 98	0.1467 ± 0.0447, 96
MML_{NU}^{WT}	0.1868 ± 0.0837, 88	0.2395 ± 0.0908, 33	0.2527 ± 0.0882, 13
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1874 ±0.0059 ^{*1}, 0	0.1882 ±1.1163 $*^{-}$ 16 ^{*8}, 0	0.1882 ±2.2335 $*^{-}$ 16 ^{*14}, 0
Lagrangian	0.1813 ± 0.0982, 0	0.1792 ± 0.0245, 0	0.1861 ± 0.0089, 0
SMOBR	0.2059 ± 0.0575, 0	0.1865 ± 0.0557, 0	0.1819 ± 0.0575, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.1965 ± 0.0917, 85	0.0615 ± 0.0449, 3	0.0345 ± 0.0282, 0
PCA-MML$_{SR}^{WT}$	0.1504 ± 0.0456, 78	0.0373 ± 0.0223, 2	0.0199 ± 0.0167, 0
PCA-MML$_{NUR}^{WT}$	0.2333 ± 0.0986, 89	0.0475 ± 0.0430, 3	0.0264 ± 0.0168, 0
PCA-MML^{ANG}	0.1081 ± 0.0737, 72	0.0293 ± 0.0242, 1	0.0243 ± 0.0186, 0
MML_{NU}^{WT}	0.1741 ± 0.0567, 62	0.1754 ± 0.1075, 0	0.0304 ± 0.0398, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1816 ± 0.0166, 0	0.1667 ±0.0230 ^{*3}, 0	0.1661 ±0.0244 ^{*4}, 0
Lagrangian	0.1341 ± 0.0783, 0	0.0510 ± 0.0322, 0	0.0316 ± 0.0152, 0
SMOBR	0.1877 ± 0.0007, 15	0.1878 ± 0.0044, 0	0.1882 ±8.0273 $*^{-}$ 6, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.3069 ± 0.0299, 98	0.3554 ± 0.0401, 96	0.3452 ± 0.0583, 97
PCA-MML$_{SR}^{WT}$	0.2957 ± 0.0878, 96	0.4443 ± N/A, 99	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.3417 ± 0.0523, 95	0.3994 ± 0.0612, 98	0.3578 ± 0.0264, 97
PCA-MML^{ANG}	0.2921 ± N/A, 99	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3272 ± 0.1087, 82	0.3521 ± 0.0920, 81	0.3142 ± 0.0552, 46
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.3294 ± 0.0838, 0	0.3170 ±0.0350 ^{*2}, 0	0.3132 ±0.0296 ^{*4}, 0
Lagrangian	0.3665 ± 0.0799, 0	0.3613 ± 0.0630, 0	0.3679 ± 0.0647, 0
SMOBR	0.1858 ± 0.0043, 0	0.1949 ± 0.0251, 0	0.2055 ± 0.0545, 0

Table 17. Area (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 10 to 100 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.2055 ± 0.1121, 69	0.1038 ± 0.0802, 10
PCA-MML$_{SR}^{WT}$	0.2066 ± 0.1198, 89	0.0817 ± 0.0553, 40
PCA-MML$_{NUR}^{WT}$	0.1906 ± 0.1191, 67	0.1042 ± 0.0937, 10
PCA-MML^{ANG}	0.2123 ± 0.1696, 97	0.0478 ± 0.0443, 32
MML_{NU}^{WT}	0.2555 ± 0.0841, 0	0.2228 ± 0.0889, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1887 ±0.0043 ^{*21}, 0	0.1882 ±1.3978 $*^{-}$ 16 ^{*30}, 0
Lagrangian	0.0971 ± 0.0492, 0	0.0716 ± 0.0384, 0
SMOBR	0.1844 ± 0.0589, 0	0.1788 ± 0.0585, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0619 ± 0.0370, 0	0.0334 ± 0.0235, 0
PCA-MML$_{SR}^{WT}$	0.0469 ± 0.0363, 0	0.0215 ± 0.0159, 0
PCA-MML$_{NUR}^{WT}$	0.0472 ± 0.0259, 0	0.0326 ± 0.0176, 0
PCA-MML^{ANG}	0.0295 ± 0.0333, 0	0.0131 ± 0.0103, 0
MML_{NU}^{WT}	0.1488 ± 0.1005, 0	0.0725 ± 0.0918, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1882 ±1.6772 $*^{-}$ 16 ^{*29}, 0	0.1882 ±5.6014 $*^{-}$ 17 ^{*44}, 0
Lagrangian	0.1882 ± 3.3474e-16, 0	0.1882 ± 3.3474e-16, 0
SMOBR	0.1665 ± 0.0255, 0	0.1727 ± 0.0359, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0127 ± 0.0101, 0	0.0087 ± 0.0069, 0
PCA-MML$_{SR}^{WT}$	0.0044 ± 0.0037, 0	0.0016 ± 0.0017, 0
PCA-MML$_{NUR}^{WT}$	0.0125 ± 0.0077, 0	0.0102 ± 0.0050, 0
PCA-MML^{ANG}	0.0072 ± 0.0070, 0	0.0042 ± 0.0035, 0
MML_{NU}^{WT}	0.0122 ± 0.0064, 0	0.0113 ± 0.0054, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.1545 ±0.0263 ^{*18}, 0	0.1570 ±0.0296 ^{*38}, 0
Lagrangian	0.0221 ± 0.0097, 0	0.0208 ± 0.0075, 0
SMOBR	0.1882 ±5.1607 $*^{-}$ 6, 0	0.1882 ±1.3655 $*^{-}$ 6, 0
50/50		
PCA-MML$_{NU}^{WT}$	0.4293 ± N/A, 99	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3162 ± 0.0496, 0	0.3177 ± 0.0407, 0
MML_{SR}^{WT}	N/A ± N/A, 100	N/A ± N/A, 100
SVM^{light}	0.3172 ±0.0232 ^{*13}, 0	0.3148 ±0.0196 ^{*30}, 0
Lagrangian	0.3793 ± 0.0659, 0	0.3659 ± 0.0601, 0
SMOBR	0.2161 ± 0.0678, 0	0.2133 ± 0.0629, 0

Table 18. Area (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Translation Matrix 1 - data sets sizes 500 to 1000 points. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

D PCA-Spikey Area Tables for Artificial Binomial Data, Translation Matrix 2

The following tables show the Area between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, two of the Spikey methods ([16, 17]) and the SVMs, SVM^{light}, the Lagrangian SVM and SMOBR for artificial Binary data skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3051 ± 0.0816, 91	0.4260 ± 0.0667, 95	0.3334 ± 0.0868, 89
PCA-MML$_{SR}^{WT}$	0.2888 ± 0.0735, 96	0.4388 ± N/A, 99	0.3731 ± 0.0995, 96
PCA-MML$_{NUR}^{WT}$	0.2747 ± 0.0644, 92	0.3584 ± 0.0729, 94	0.2995 ± 0.0648, 89
PCA-MML^{ANG}	0.2869 ± 0.0215, 97	N/A ± N/A, 100	0.4913 ± N/A, 99
MML_{NU}^{WT}	0.2720 ± 0.0988, 79	0.2628 ± 0.0781, 51	0.2613 ± 0.0841, 21
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2413 ±0.0445 ^{*2}, 0	0.2355 ±0.0275 ^{*7}, 0	0.2365 ±0.0223 ^{*8}, 0
Lagrangian	0.2667 ± 0.1104, 0	0.2077 ± 0.1020, 0	0.1878 ± 0.0922, 0
SMOBR	0.2282 ± 0.0950, 0	0.2580 ± 0.1090, 0	0.2391 ± 0.0970, 0
85/55			
PCA-MML$_{NU}^{WT}$	0.3146 ± 0.0806, 89	0.2777 ± 0.1038, 85	0.2478 ± 0.0965, 74
PCA-MML$_{SR}^{WT}$	0.3675 ± 0.0954, 93	0.2808 ± 0.0754, 95	0.2701 ± 0.1340, 90
PCA-MML$_{NUR}^{WT}$	0.3256 ± 0.0962, 89	0.2854 ± 0.0854, 85	0.2618 ± 0.0914, 75
PCA-MML^{ANG}	0.3730 ± 0.0617, 95	0.3125 ± 0.1292, 93	0.2878 ± 0.1274, 95
MML_{NU}^{WT}	0.2596 ± 0.0972, 74	0.2566 ± 0.0778, 34	0.2333 ± 0.0895, 4
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2989 ±0.0759 ^{*5}, 0	0.2655 ±0.0440 ^{*1}, 0	0.2587 ±0.0279 ^{*1}, 0
Lagrangian	0.3681 ± 0.0899, 0	0.4066 ± 0.0769, 0	0.3998 ± 0.0806, 0
SMOBR	0.2475 ± 0.0952, 0	0.2572 ± 0.0968, 0	0.2367 ± 0.0756, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.2183 ± 0.0918, 73	0.0585 ± 0.0352, 0	0.0411 ± 0.0239, 0
PCA-MML$_{SR}^{WT}$	0.1591 ± 0.0634, 74	0.0418 ± 0.0278, 0	0.0244 ± 0.0217, 0
PCA-MML$_{NUR}^{WT}$	0.2036 ± 0.0603, 79	0.0555 ± 0.0395, 0	0.0404 ± 0.0233, 0
PCA-MML^{ANG}	0.2878 ± 0.1274, 66	0.0301 ± 0.0246, 0	0.0370 ± 0.0286, 0
MML_{NU}^{WT}	0.1818 ± 0.0671, 46	0.0764 ± 0.0467, 0	0.0620 ± 0.0233, 0
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2272 ± 0.0410, 0	0.1961 ±0.0449 ^{*2}, 0	0.1649 ± 0.0347, 0
Lagrangian	0.0547 ± 0.0516, 0	0.0467 ± 0.0269, 0	0.0329 ± 0.0179, 0
SMOBR	0.1562 ± 0.0923, 0	0.0754 ± 0.0551, 0	0.0739 ± 0.0510, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.2789 ± N/A, 99	0.3743 ± 0.0734, 97	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	0.4234 ± 0.0229, 98	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.3238 ± 0.0943, 96	0.4393 ± 0.0705, 96	N/A ± N/A, 100
PCA-MML^{ANG}	0.2832 ± N/A, 99	0.4199 ± N/A, 99	N/A ± N/A, 100
MML_{NU}^{WT}	0.3111 ± 0.1132, 80	0.3036 ± 0.0903, 64	0.2778 ± 0.0576, 45
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2841 ±0.0804 ^{*1}, 0	0.2683 ± 0.0401, 0	0.2613 ±0.0227 ^{*2}, 0
Lagrangian	0.3427 ± 0.0968, 0	0.3521 ± 0.0863, 0	0.3348 ± 0.0771, 0
SMOBR	0.2227 ± 0.0819, 0	0.2609 ± 0.0791, 0	0.2557 ± 0.0627, 0

Table 19. Area (mean ± standard deviation) Between True Hyperplane ($y = 1.6x + 10.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 10 to 100. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.1648 ± 0.1161, 43	0.0612 ± 0.0652, 1
PCA-MML$_{SR}^{WT}$	0.1869 ± 0.1231, 79	0.0764 ± 0.0639, 12
PCA-MML$_{NUR}^{WT}$	0.2270 ± 0.1115, 54	0.1063 ± 0.0709, 3
PCA-MML^{ANG}	0.2179 ± 0.1567, 95	0.0407 ± 0.0421, 2
MML_{NU}^{WT}	0.1815 ± 0.0939, 0	0.0930 ± 0.0743, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.2364 ±0.0273 ^{*16}, 0	0.2447 ±0.0272 ^{*18}, 0
Lagrangian	0.0930 ± 0.0496, 0	0.0630 ± 0.0399, 0
SMOBR	0.2119 ± 0.0976, 0	0.1993 ± 0.0784, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0482 ± 0.0555, 0	0.0224 ± 0.0203, 0
PCA-MML$_{SR}^{WT}$	0.0457 ± 0.0493, 0	0.0170 ± 0.0200, 0
PCA-MML$_{NUR}^{WT}$	0.0802 ± 0.0571, 0	0.0457 ± 0.0187, 0
PCA-MML^{ANG}	0.0363 ± 0.0415, 0	0.0141 ± 0.0122, 0
MML_{NU}^{WT}	0.0686 ± 0.0528, 0	0.0512 ± 0.0135, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.2531 ±4.4642 $*^{-}$ 16 ^{*4}, 0	0.2531 ±3.9068 $*^{-}$ 16 ^{*7}, 0
Lagrangian	0.3774 ± 0.0395, 0	0.3669 ± 0.0306, 0
SMOBR	0.2270 ± 0.0240, 0	0.2424 ± 0.0398, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0148 ± 0.0117, 0	0.0128 ± 0.0123, 0
PCA-MML$_{SR}^{WT}$	0.0046 ± 0.0062, 0	0.0021 ± 0.0019, 0
PCA-MML$_{NUR}^{WT}$	0.0275 ± 0.0106, 0	0.0308 ± 0.0073, 0
PCA-MML^{ANG}	0.0089 ± 0.0065, 0	0.0048 ± 0.0052, 0
MML_{NU}^{WT}	0.0496 ± 0.0065, 0	0.0488 ± 0.0046, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.1068 ±0.0079 ^{*1}, 0	0.0780 ±0.0091 ^{*2}, 0
Lagrangian	0.0164 ± 0.0085, 0	0.0120 ± 0.0066, 0
SMOBR	0.0819 ± 0.0336, 0	0.0834 ± 0.0223, 0
50/50		
PCA-MML$_{NU}^{WT}$	0.3503 ± 0.0710, 97	0.2533 ± N/A, 99
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.2575 ± 0.0056, 98	0.2553 ± N/A, 99
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.2553 ± 0.0616, 1	0.2574 ± 0.0387, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.2590 ±0.0207 ^{*11}, 0	0.2624 ±0.0121 ^{*11}, 0
Lagrangian	0.3536 ± 0.0822, 0	0.3526 ± 0.0815, 0
SMOBR	0.2487 ± 0.0614, 0	0.2546 ± 0.0226, 0

Table 20. Area (mean ± standard deviation) Between True Hyperplane ($y = 1.6x + 10.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 500 to 1000. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3524 ± 0.0937, 86	0.3264 ± 0.0736, 94	0.2947 ± 0.0599, 97
PCA-MML$_{SR}^{WT}$	0.3099 ± 0.1058, 96	0.4307 ± N/A, 99	0.3475 ± N/A, 99
PCA-MML$_{NUR}^{WT}$	0.2982 ± 0.0747, 91	0.3228 ± 0.0130, 95	0.3481 ± 0.0499, 95
PCA-MML^{ANG}	0.2845 ± 0.1025, 95	N/A ± N/A, 100	0.4100 ± N/A, 99
MML_{NU}^{WT}	0.3422 ± 0.0831, 84	0.3336 ± 0.0685, 49	0.3312 ± 0.0585, 32
MML_{SR}^{WT}	0.2000 ± 3.103e-17, 95	NA, 100	NA, 100
SVM^{light}	0.2631 ±0.0697 ^{*1}, 0	0.2239 ±0.0493 ^{*2}, 0	0.2139 ±0.0290 ^{*3}, 0
Lagrangian	0.2864 ± 0.1092, 0	0.2160 ± 0.1021, 0	0.1635 ± 0.0752, 0
SMOBR	0.2555 ± 0.0887, 0	0.2381 ± 0.0698, 0	0.2206 ± 0.0598, 0
85/55			
PCA-MML$_{NU}^{WT}$	0.3201 ± 0.0700, 86	0.3045 ± 0.0874, 83	0.3212 ± 0.0585, 86
PCA-MML$_{SR}^{WT}$	0.3147 ± 0.0941, 90	0.3228 ± 0.0862, 97	0.2599 ± 0.0960, 95
PCA-MML$_{NUR}^{WT}$	0.3227 ± 0.0675, 84	0.3142 ± 0.0958, 88	0.3044 ± 0.1102, 85
PCA-MML^{ANG}	0.3370 ± 0.0925, 91	0.3978 ± N/A, 99	0.2761 ± N/A, 99
MML_{NU}^{WT}	0.3302 ± 0.0757, 77	0.3106 ± 0.0790, 43	0.3230 ± 0.0636, 18
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2831 ± 0.0942, 0	0.2139 ±0.0554 ^{*2}, 0	0.2027 ±0.0243 ^{*5}, 0
Lagrangian	0.3728 ± 0.0829, 0	0.3843 ± 0.0903, 0	0.3578 ± 0.0873, 0
SMOBR	0.2704 ± 0.1009, 0	0.2418 ± 0.0788, 0	0.2674 ± 0.0848, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.2154 ± 0.1047, 78	0.0494 ± 0.0445, 1	0.0331 ± 0.0249, 0
PCA-MML$_{SR}^{WT}$	0.1799 ± 0.0808, 74	0.0258 ± 0.0278, 1	0.0151 ± 0.0128, 0
PCA-MML$_{NUR}^{WT}$	0.2517 ± 0.0955, 83	0.0409 ± 0.0306, 1	0.0262 ± 0.0173, 0
PCA-MML^{ANG}	0.1141 ± 0.0770, 65	0.0248 ± 0.0206, 1	0.0349 ± 0.0310, 0
MML_{NU}^{WT}	0.2984 ± 0.0869, 69	0.2520 ± 0.0806, 3	0.2054 ± 0.1049, 0
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.2114 ± 0.0319, 0	0.1958 ±0.0092 ^{*2}, 0	0.1948 ±0.0122 ^{*2}, 0
Lagrangian	0.1199 ± 0.0975, 0	0.0443 ± 0.0343, 0	0.0292 ± 0.0181, 0
SMOBR	0.2008 ± 0.0054, 13	0.2016 ± 0.0150, 0	0.2001 ±6.6543 $*^{-}$ 6, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.3727 ± 0.0611, 98	0.3546 ± 0.0991, 95	0.4811 ± N/A, 99
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100	0.2719 ± N/A, 99
PCA-MML$_{NUR}^{WT}$	0.3196 ± 0.0307, 98	0.3565 ± 0.1002, 95	N/A ± N/A, 100
PCA-MML^{ANG}	0.4239 ± N/A, 99	0.4355 ± N/A, 99	N/A ± N/A, 100
MML_{NU}^{WT}	0.3161 ± 0.0617, 81	0.3141 ± 0.0724, 74	0.3272 ± 0.0489, 30
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.3279 ±0.0714 ^{*1}, 0	0.3272 ±0.0344 ^{*1}, 0	0.3247 ±0.0207 ^{*2}, 0
Lagrangian	0.3505 ± 0.0951, 0	0.3508 ± 0.0617, 0	0.3649 ± 0.0632, 0
SMOBR	0.2812 ± 0.1066, 0	0.2698 ± 0.0950, 0	0.2754 ± 0.0940, 0

Table 21. Area (mean ± standard deviation) Between True Hyperplane ($y = -0.9x + 130.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 10 to 100. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.1762 ± 0.1248, 68	0.0666 ± 0.0717, 1
PCA-MML$_{SR}^{WT}$	0.1573 ± 0.0892, 89	0.0349 ± 0.0425, 0
PCA-MML$_{NUR}^{WT}$	0.1802 ± 0.1245, 62	0.0721 ± 0.0805, 1
PCA-MML^{ANG}	0.1037 ± 0.0328, 96	0.0562 ± 0.0490, 27
MML_{NU}^{WT}	0.3013 ± 0.0719, 0	0.2867 ± 0.0782, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.1993 ± 0.0074, 0	0.2000 ± 1.9544e-16 ^{*14}, 0
Lagrangian	0.0867 ± 0.0463, 0	0.0681 ± 0.0413, 0
SMOBR	0.2216 ± 0.0636, 0	0.2441 ± 0.0872, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0440 ± 0.0460, 0	0.0249 ± 0.0355, 0
PCA-MML$_{SR}^{WT}$	0.0265 ± 0.0345, 0	0.0110 ± 0.0117, 0
PCA-MML$_{NUR}^{WT}$	0.0459 ± 0.0496, 0	0.0171 ± 0.0184, 0
PCA-MML^{ANG}	0.0365 ± 0.0311, 1	0.0165 ± 0.0125, 0
MML_{NU}^{WT}	0.2895 ± 0.0841, 0	0.1854 ± 0.1003, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.2000 ±3.9064 $*^{-}$ 16 ^{*5}, 0	0.2000 ±3.9058 $*^{-}$ 16 ^{*2}, 0
Lagrangian	0.3205 ± 0.0474, 0	0.3139 ± 0.0358, 0
SMOBR	0.2303 ± 0.0463, 0	0.2070 ± 0.0070, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0172 ± 0.0130, 0	0.0154 ± 0.0100, 0
PCA-MML$_{SR}^{WT}$	0.0047 ± 0.0038, 0	0.0024 ± 0.0017, 0
PCA-MML$_{NUR}^{WT}$	0.0139 ± 0.0109, 0	0.0104 ± 0.0076, 0
PCA-MML^{ANG}	0.0099 ± 0.0079, 0	0.0059 ± 0.0062, 0
MML_{NU}^{WT}	0.0992 ± 0.0252, 0	0.0927 ± 0.0162, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.1620 ±0.0432 ^{*8}, 0	0.0843 ±0.0273 ^{*13}, 0
Lagrangian	0.0144 ± 0.0083, 0	0.0123 ± 0.0064, 0
SMOBR	0.2001 ±8.0422 $*^{-}$ 6, 0	0.2001 ±4.6547 $*^{-}$ 6, 0
50/50		
PCA-MML$_{NU}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3309 ± 0.0313, 0	0.3244 ± 0.0430, 0
MML_{SR}^{WT}NA, 100	NA, 100	
SVM^{light}	0.3264 ±0.0290 ^{*5}, 0	0.3238 ±0.0195 ^{*7}, 0
Lagrangian	0.3644 ± 0.0688, 0	0.3695 ± 0.0679, 0
SMOBR	0.2577 ± 0.0868, 0	0.2549 ± 0.0878, 0

Table 22. Area (mean ± standard deviation) Between True Hyperplane ($y = -0.9x + 130.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 500 to 1000. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 10	**N = 50**	**N = 100**
60/40			
PCA-MML$_{NU}^{WT}$	0.3704 ± 0.0896, 90	0.2211 ± 0.0489, 94	0.3139 ± 0.0110, 96
PCA-MML$_{SR}^{WT}$	0.1625 ± 0.0240, 98	0.4510 ± N/A, 99	0.3076 ± 0.0172, 98
PCA-MML$_{NUR}^{WT}$	0.3375 ± 0.0741, 94	0.3099 ± 0.1764, 97	0.3023 ± 0.0343, 95
PCA-MML^{ANG}	0.2360 ± 0.1522, 98	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3212 ± 0.1099, 77	0.2859 ± 0.0998, 44	0.3118 ± 0.0587, 25
MML_{SR}^{WT}	0.8118 ± 1.2413e-16, 95	NA, 100	
SVM^{light}	0.2251 ± 0.0711, 0	0.1956 ± 0.0238, 0	0.1923 ±0.0208 ^{*2}, 0
Lagrangian	0.2975 ± 0.1073, 0	0.2253 ± 0.0980, 0	0.1710 ± 0.0912, 0
SMOBR	0.2174 ± 0.0660, 0	0.2070 ± 0.0651, 0	0.1888 ± 0.0188, 0
85/55			
PCA-MML$_{NU}^{WT}$	N/A ± N/A, 100	0.1745 ± 0.0644, 91	0.2501 ± 0.1151, 82
PCA-MML$_{SR}^{WT}$	0.35118 ± 0.0976, 98	0.1548 ± 0.0314, 91	0.1779 ± 0.1012, 90
PCA-MML$_{NUR}^{WT}$	0.4932 ± N/A, 99	0.1954 ± 0.0913, 88	0.2256 ± 0.1258, 80
PCA-MML^{ANG}	0.1707 ± N/A, 99	0.1843 ± 0.1221, 98	0.1467 ± 0.0447, 96
MML_{NU}^{WT}	0.1811 ± 0.0917, 88	0.2291 ± 0.0945, 31	0.2496 ± 0.1018, 7
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.1875 ±0.0061 ^{*2}, 0	0.1882 ± 3.3476e-16 ^{*1}, 0	0.1882 ± 3.3476e-16 ^{*1}, 0
Lagrangian	0.1807 ± 0.0982, 0	0.1792 ± 0.0245, 0	0.1861 ± 0.0089, 0
SMOBR	0.2044 ± 0.0642, 0	0.1929 ± 0.0539, 0	0.1826 ± 0.0525, 0
95/05			
PCA-MML$_{NU}^{WT}$	0.1807 ± 0.0932, 82	0.0577 ± 0.0458, 2	0.0304 ± 0.0248, 0
PCA-MML$_{SR}^{WT}$	0.1447 ± 0.0386, 79	0.0318 ± 0.0205, 3	0.0202 ± 0.0142, 0
PCA-MML$_{NUR}^{WT}$	0.2187 ± 0.1037, 89	0.0491 ± 0.0454, 2	0.0253 ± 0.0140, 0
PCA-MML^{ANG}	0.1102 ± 0.0705, 68	0.0292 ± 0.0242, 1	0.0246 ± 0.0189, 0
MML_{NU}^{WT}	0.1738 ± 0.068, 57	0.1553 ± 0.0992, 0	0.0483 ± 0.0337, 0
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.1788 ± 0.0211, 0	0.1569 ±0.0313 ^{*1}, 0	0.1500 ±0.0349 ^{*1}, 0
Lagrangian	0.1366 ± 0.0821, 0	0.0510 ± 0.0322, 0	0.0316 ± 0.0152, 0
SMOBR	0.1882 ± 0.0000, 15	0.1895 ± 0.0152, 0	0.1882 ±2.4042 $*^{-}$ 6, 0
50/50			
PCA-MML$_{NU}^{WT}$	0.3371 ± 0.0231, 97	0.3567 ± 0.0370, 96	0.3445 ± 0.0640, 97
PCA-MML$_{SR}^{WT}$	0.3080 ± 0.0788, 95	0.4482 ± N/A, 99	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.3244 ± 0.0502, 93	0.3735 ± 0.0281, 97	0.3780 ± N/A, 99
PCA-MML^{ANG}	0.2921 ± N/A, 99	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3311 ± 0.0979, 78	0.3430 ± 0.0760, 78	0.3211 ± 0.0626, 37
MML_{SR}^{WT}	NA, 100	NA, 100	NA, 100
SVM^{light}	0.3264 ± 0.0804, 0	0.3211 ± 0.0458, 0	0.3093 ± 0.0289, 0
Lagrangian	0.3665 ± 0.0799, 0	0.3613 ± 0.0630, 0	0.3679 ± 0.0647, 0
SMOBR	0.2269 ± 0.0775, 0	0.2208 ± 0.0708, 0	0.2171 ± 0.0606, 0

Table 23. Area (mean ± standard deviation) Between True Hyperplane ($y = -1.7x + 80.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 10 to 100. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

	N = 500	**N = 1000**
60/40		
PCA-MML$_{NU}^{WT}$	0.1924 ± 0.1147, 71	0.1079 ± 0.0871, 11
PCA-MML$_{SR}^{WT}$	0.1914 ± 0.1183, 87	0.0790 ± 0.0459, 43
PCA-MML$_{NUR}^{WT}$	0.2037 ± 0.1185, 70	0.0963 ± 0.0890, 8
PCA-MML^{ANG}	0.2123 ± 0.1696, 97	0.0475 ± 0.0443, 32
MML_{NU}^{WT}	0.2279 ± 0.0903, 0	0.1971 ± 0.0992, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.1886 ±0.0038 ^{*4}, 0	0.1882 ± 3.0700e-16 ^{*9}, 0
Lagrangian	0.0971 ± 0.0492, 0	0.0716 ± 0.0384, 0
SMOBR	0.1800 ± 0.0376, 0	0.1829 ± 0.0652, 0
85/55		
PCA-MML$_{NU}^{WT}$	0.0557 ± 0.0484, 0	0.0281 ± 0.0224, 0
PCA-MML$_{SR}^{WT}$	0.0421 ± 0.0377, 0	0.0184 ± 0.0131, 0
PCA-MML$_{NUR}^{WT}$	0.0473 ± 0.0314, 0	0.0315 ± 0.0135, 0
PCA-MML^{ANG}	0.0296 ± 0.0333, 0	0.0130 ± 0.0103, 0
MML_{NU}^{WT}	0.1366 ± 0.0827, 0	0.0557 ± 0.0365, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.1882 ± 3.3480e-16 ^{*3}, 0	0.1882 ± 3.3480e-16 ^{*3}, 0
Lagrangian	0.1882 ±2.7895 $*^{-}$ 17, 0	0.1882 ±2.7895 $*^{-}$ 17, 0
SMOBR	0.1630 ± 0.0549, 0	0.1661 ± 0.0412, 0
95/05		
PCA-MML$_{NU}^{WT}$	0.0122 ± 0.0101, 0	0.0086 ± 0.0069, 0
PCA-MML$_{SR}^{WT}$	0.0032 ± 0.0027, 0	0.0017 ± 0.0012, 0
PCA-MML$_{NUR}^{WT}$	0.0120 ± 0.0078, 0	0.0106 ± 0.0059, 0
PCA-MML^{ANG}	0.0072 ± 0.0070, 0	0.0042 ± 0.0035, 0
MML_{NU}^{WT}	0.0210 ± 0.0054, 0	0.0183 ± 0.0026, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.0908 ±0.0290 ^{*11}, 0	0.0528 ±0.0082 ^{*20}, 0
Lagrangian	0.0221 ± 0.0097, 0	0.0208 ± 0.0075, 0
SMOBR	0.1882 ±1.3027 $*^{-}$ 6, 0	0.1882 ±1.6476 $*^{-}$ 6, 0
50/50		
PCA-MML$_{NU}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML^{ANG}	N/A ± N/A, 100	N/A ± N/A, 100
MML_{NU}^{WT}	0.3138 ± 0.0358, 0	0.3150 ± 0.0351, 0
MML_{SR}^{WT}	NA, 100	NA, 100
SVM^{light}	0.3143 ±0.0196 ^{*3}, 0	0.3124 ±0.0224 ^{*3}, 0
Lagrangian	0.3793 ± 0.0660, 0	0.3658 ± 0.0600, 0
SMOBR	0.2282 ± 0.0790, 0	0.2114 ± 0.0618, 0

Table 24. Area (mean ± standard deviation) Between True Hyperplane ($y = -1.7x + 80.0$) and Inferred Hyperplanes for Translation Matrix 2 - data set sizes 500 to 1000. The 'n' in *mean* ± *standard deviation* ^{*n} denotes the number of data sets on which SVM^{light} did not converge.

PCA-Spikey Results Tables for Trinomial Data - Kullback-Leibler Distance and Area Between True and Inferred Hyperplanes

E PCA-Spikey Kullback-Leibler Distance Tables for Artificial Trinomial Data, (95%/05%/05%/95%)

The following tables show the Kullback-Leibler distances between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (95%/05%/05%/95%) distribution and skewed via Translation Matrix 1.

	N = 10	**N = 50**	**N = 100**
95/05/05/95			
PCA-MML$_{NU}^{WT}$	0.2681 ± 0.0702	0.1156 ± 0.0553	0.0819 ± 0.0449
PCA-MML$_{SR}^{WT}$	0.2779 ± 0.0686	0.0948 ± 0.0528	0.0501 ± 0.0429
PCA-MML$_{NUR}^{WT}$	0.2781 ± 0.0639	0.1109 ± 0.0568	0.0834 ± 0.0415
PCA-MML^{ANG}	0.2800 ± 0.0604	0.3077 ± 0.0778	0.3174 ± 0.0528

Table 25. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (95/05/05/95) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
95/05/05/95		
PCA-MML$_{NU}^{WT}$	0.0352 ± 0.0233	0.0303 ± 0.0240
PCA-MML$_{SR}^{WT}$	0.0103 ± 0.0087	0.0057 ± 0.0049
PCA-MML$_{NUR}^{WT}$	0.0620 ± 0.0201	0.0683 ± 0.0136
PCA-MML^{ANG}	0.3142 ± 0.0744	0.3042 ± 0.0866

Table 26. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (95/05/05/95) - data sets sizes 500 to 1000 points.

F PCA-Spikey Kullback-Leibler Distance Tables for Artificial Trinomial Data, (60%/30%/10%/70%)

The following tables show the Kullback-Leibler distances between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (60%/30%/10%/70%) distribution and skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
60/30/10/70			
PCA-MML$_{NU}^{WT}$	0.1125 ± 0.0484	0.0388 ± 0.0341	0.0199 ± 0.0113
PCA-MML$_{SR}^{WT}$	0.1129 ± 0.0488	0.0361 ± 0.0250	0.0199 ± 0.0124
PCA-MML$_{NUR}^{WT}$	0.1088 ± 0.0430	0.0350 ± 0.0222	0.0210 ± 0.0139
PCA-MML^{ANG}	0.1066 ± 0.0451	0.0349 ± 0.0213	0.0194 ± 0.0107

Table 27. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (60/30/10/70) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
60/30/10/70		
PCA-MML$_{NU}^{WT}$	0.0042 ± 0.0024	0.0019 ± 0.0009
PCA-MML$_{SR}^{WT}$	0.0042 ± 0.0024	0.0019 ± 0.0009
PCA-MML$_{NUR}^{WT}$	0.0041 ± 0.0021	0.0019 ± 0.0009
PCA-MML^{ANG}	0.0040 ± 0.0017	0.0019 ± 0.0009

Table 28. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (60/30/10/70) - data sets sizes 500 to 1000 points.

G PCA-Spikey Kullback-Leibler Distance Tables for Artificial Trinomial Data, (35%/35%/40%/20%)

The following tables show the Kullback-Leibler distances between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (35%/35%/40%/20%) distribution and skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
35/35/40/20			
PCA-MML$_{NU}^{WT}$	0.0968 ± 0.0472	0.0378 ± 0.0279	0.0194 ± 0.0110
PCA-MML$_{SR}^{WT}$	0.0935 ± 0.0400	0.0343 ± 0.0215	0.0186 ± 0.0102
PCA-MML$_{NUR}^{WT}$	0.0958 ± 0.0441	0.0375 ± 0.0268	0.0194 ± 0.0116
PCA-MML^{ANG}	0.0874 ± 0.0308	0.0313 ± 0.0154	0.0181 ± 0.0086

Table 29. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Trinomial Data (35/35/40/20) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
35/35/40/20		
PCA-MML$_{NU}^{WT}$	0.0039 ± 0.0022	0.0019 ± 0.0008
PCA-MML$_{SR}^{WT}$	0.0038 ± 0.0020	0.0019 ± 0.0008
PCA-MML$_{NUR}^{WT}$	0.0038 ± 0.0020	0.0019 ± 0.0008
PCA-MML^{ANG}	0.0038 ± 0.0020	0.0019 ± 0.0008

Table 30. Kullback-Leibler distances (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Trinomial Data (35/35/40/20) - data sets sizes 500 to 1000 points.

H PCA-Spikey Area Tables for Artificial Trinomial Data, (95%/05%/05%/95%)

The following tables show the areas between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (95%/05%/05%/95%) distribution and skewed via Translation Matrix 1.

	N = 10	**N = 50**	**N = 100**
95/05/05/95			
PCA-MML$_{NU}^{WT}$	0.2030 ± 0.0880, 70	0.0617 ± 0.0387, 0	0.0403 ± 0.0247, 0
PCA-MML$_{SR}^{WT}$	0.1656 ± 0.0483, 77	0.0434 ± 0.0264, 0	0.0225 ± 0.0216, 0
PCA-MML$_{NUR}^{WT}$	0.2010 ± 0.0605, 77	0.0558 ± 0.0391, 0	0.0399 ± 0.0217, 0

Table 31. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (95/05/05/95) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
95/05/05/95		
PCA-MML$_{NU}^{WT}$	0.0152 ± 0.0108, 0	0.0128 ± 0.0111, 0
PCA-MML$_{SR}^{WT}$	0.0040 ± 0.0035, 0	0.0022 ± 0.0019, 0
PCA-MML$_{NUR}^{WT}$	0.0282 ± 0.0112, 0	0.0315 ± 0.0085, 0

Table 32. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (95/05/05/95) - data sets sizes 500 to 1000 points.

I PCA-Spikey Area Tables for Artificial Trinomial Data, (60%/30%/10%/70%)

The following tables show the areas between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (60%/30%/10%/70%) distribution and skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
60/30/10/70			
PCA-MML$_{NU}^{WT}$	0.3677 ± 0.1001, 95	0.3981 ± 0.0883, 96	0.2826 ± 0.0692, 97
PCA-MML$_{SR}^{WT}$	0.3293 ± 0.1343, 91	0.3674 ± 0.1294, 97	0.2983 ± N/A, 99
PCA-MML$_{NUR}^{WT}$	0.2968 ± 0.1256, 95	0.3191 ± 0.1868, 96	0.3521 ± 0.1030, 96

Table 33. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (60/30/10/70) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
60/30/10/70		
PCA-MML$_{NU}^{WT}$	0.3206 ± 0.2226, 98	N/A ± N/A, 100
PCA-MML$_{SR}^{WT}$	0.2364 ± 0.0253, 97	N/A ± N/A, 100
PCA-MML$_{NUR}^{WT}$	0.2526 ± 0.1271, 98	N/A ± N/A, 100

Table 34. Area (mean ± standard deviation) between the true hyperplane ($y = 1.6x + 10.0$) and inferred hyperplanes for Trinomial Data (60/30/10/70) - data sets sizes 500 to 1000 points.

J PCA-Spikey Area Tables for Artificial Trinomial Data, (35%/35%/40%/20%)

The following tables show the areas between the true hyperplanes and those inferred by the four PCA-Spikey methods, as presented in Section 6, for artificial Trinomial data with a (35%/35%/40%/20%) distribution and skewed via Translation Matrix 2.

	N = 10	**N = 50**	**N = 100**
35/35/40/20			
PCA-MML$^{WT}_{NU}$	0.3625 ± 0.0980, 91	0.3212 ± 0.1203, 91	0.3657 ± 0.0504, 96
PCA-MML$^{WT}_{SR}$	0.3432 ± 0.1272, 91	0.2502 ± 0.1378, 96	0.4781 ± N/A, 99
PCA-MML$^{WT}_{NUR}$	0.3589 ± 0.070, 87	0.3216 ± 0.1142, 91	0.3764 ± 0.0570, 97

Table 35. Area (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Trinomial Data (35/35/40/20) - data sets sizes 10 to 100 points.

	N = 500	**N = 1000**
35/35/40/20		
PCA-MML$^{WT}_{NU}$	0.3952 ± N/A, 99	N/A ± N/A, 100
PCA-MML$^{WT}_{SR}$	N/A ± N/A, 100	N/A ± N/A, 100
PCA-MML$^{WT}_{NUR}$	N/A ± N/A, 100	N/A ± N/A, 100

Table 36. Area (mean ± standard deviation) between the true hyperplane ($y = -1.7x + 80.0$) and inferred hyperplanes for Trinomial Data (35/35/40/20) - data sets sizes 500 to 1000 points.

Results Tables for Real Data - Wisconsin Prognostic Breast Cancer Database and Iris Dataset. Probabilistic Prediction Bit Score Error and Right/Wrong Predictive Accuracy

K Result Table for Real Binomial Data

The following table shows the Probabilistic Prediction and Right/Wrong Predictive Accuracy scores for the Spikey, PCA-Spikey and SVM methods for Binary data obtained from the Wisconsin Prognostic Breast Cancer Database [2].

	Prob. Prediction (bit score) error	**Right/Wrong Accuracy**
MML_{NU}^{WT}	10.035377 ± 4.689809	0.960580 ± 0.023718
MML_{SR}^{WT}	44.540905 ± 2.579899	0.655072 ± 0.059755
PCA-MML_{NU}^{WT}	14.0531 ± 13.6865	0.905714 ± 0.160809
PCA-MML_{SR}^{WT}	10.1341 ± 7.52156	0.957143 ± 0.0349927
PCA-MML_{NUR}^{WT}	9.97667 ± 7.52561	0.957143 ± 0.0368856
Lagrangian	12.029075 ± 6.396449	0.958261 ± 0.028850
SMOBR	44.540905 ± 2.579899	0.655072 ± 0.059755

Table 37. Real Data Set - Wisconsin Prognostic Breast Cancer Database. Probabilistic prediction bit score error results and "right/wrong" predictive accuracy results using 10-fold cross-validation (mean ± standard deviation).

L PCA-Spikey Result Table for Real Trinomial Data (2D)

The following table shows the Probabilistic Prediction and Right/Wrong Predictive Accuracy scores for the PCA-Spikey methods for two-dimensional Trinomial data obtained from the Iris data set [11].

	Prob. Prediction (bit score) error	**Right/Wrong Accuracy**
PCA-MML$_{NU}^{WT}$	8.2333 ± 1.5605	0.6567 ± 0.0568
PCA-MML$_{SR}^{WT}$	7.8656 ± 1.1779	0.6667 ± 0.0544
PCA-MML$_{NUR}^{WT}$	7.3257 ± 1.6426	0.6667 ± 0.0685
PCA-MML^{ANG}	8.0853 ± 1.6950	0.6600 ± 0.0717

Table 38. Real Data Set - Iris, 2D. Probabilistic prediction bit score error results and "right/wrong" predictive accuracy results using 10-fold cross-validation (mean ± standard deviation).

M PCA-Spikey Result Table for Real Trinomial Data (4D)

The following table shows the Probabilistic Prediction and Right/Wrong Predictive Accuracy scores for the PCA-Spikey methods (Section 6) for four-dimensional Trinomial data obtained from the Iris data set [11].

	Prob. Prediction (bit score) error	**Right/Wrong Accuracy**
PCA-MML$_{NU}^{WT}$	11.9636 ± 2.4283	0.5500 ± 0.0972
PCA-MML$_{SR}^{WT}$	10.2483 ± 1.3722	0.5767 ± 0.0686
PCA-MML$_{NUR}^{WT}$	10.0826 ± 1.6478	0.5933 ± 0.0625

Table 39. Real Data Set - Iris, 4D. Probabilistic prediction bit score error results and "right/wrong" predictive accuracy results using 10-fold cross-validation (mean ± standard deviation).

References

1. Agusta, Y. and Dowe, D. L. (2003). Unsupervised Learning of Correlated Multivariate Gaussian Mixture Models using MML, Proc. 16th Australian Joint Conference on Artificial Intelligence (AI'03), Perth, Australia, 3-5 Dec. 2003, Published in Lecture Notes in Artificial Intelligence (LNAI) 2903, Springer-Verlag, pp477-489.
2. Blake, C.L. and Merz, C.J. (1998). *UCI Repository of machine learning databases*. http://www.ics.uci.edu/∼mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.
3. Burges, Christopher J.C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, Volume 2, Pages 121-167.
4. Collie, M.J. and Dowe, D.L. and Fitzgibbon, L.J. (2005). Stock Market Simulation and Inference Technique, accepted and in press, Fifth International Conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro - Brazil, November 06-09, 2005
5. Comley, J.W. and Dowe, D.L. (2003). General Bayesian Networks and Asymmetric Languages, in Proc. Hawaii International Conference on Statistics and Related Fields, 5-8 June, 2003.
6. Comley, J.W. and Dowe, D.L. (2005). Minimum Message Length, MDL and Generalised Bayesian Networks with Asymmetric Languages, Chapter 11 in P. Grunwald, I. J. Myung and M. A. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, M.I.T. Press, April 2005, ISBN 0-262-07262-9. Final camera-ready copy submitted in October 2003.
7. Dowe, D.L. and Farr, G.E. and Hurst, A.J. and Lentin, K.L. (1996). Information-theoretic football tipping, in N. de Mestre (ed.), Third Australian Conference on Mathematics and Computers in Sport, Bond University, Qld, 233-241, 1996.

8. Dowe, D.L and Krusel, N. (1993). A decision tree model of bushfire activity, (Technical report 93/190) Dept Computer Science, Monash University, Melbourne, 7pp, 1993
9. Dowe, D.L. and Oliver, J.J. and Wallace, C.S. (1996). MML estimation of the parameters of the spherical Fisher Distribution. Proc. 7th International Workshop on Algorithmic Learning Theory (ALT'96), Lecture Notes in Artificial Intelligence (LNAI) 1160, pp213-227, Sydney, Australia, 23-25 October 1996.
10. Dowe, D.L. and Wallace, C.S. (1998). Kolmogorov complexity, minimum message length and inverse learning, abstract, page 144, 14th Australian Statistical Conference (ASC-14), Gold Coast, Qld, 6-10 July, 1998.
11. Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems, in Annals of Eugenics, Volume 7, pp. 179-188.
12. Fitzgibbon, L.J. and Dowe, D.L. and Allison, L. (2002). Univariate Polynomial Inference by Monte Carlo Message Length Approximations, Proc. 19th International Conference on Machine Learning (ICML-2002), C.Sammut and A.Hoffmann (eds.), San Francisco, Morgan Kaufmann. University of NSW, Sydney, Australia. pp 147-154, Vol 2417, ISSN:0302-9743.
13. Fitzgibbon, L.J. and Dowe, D. L. and Vahid, F. (2004). Minimum Message Length Autoregressive Model Order Selection. In M. Palanaswami, C. Chandra Sekhar, G. Kumar Venayagamoorthy, S. Mohan and M. K. Ghantasala (eds.), International Conference on Intelligent Sensing and Information Processing (ICISIP), Chennai, India, 4-7 January 2004 (ISBN: 0-7803-8243-9, IEEE Catalogue Number: 04EX783), pp439-444.
14. Joachims, T. (1998). Making Large-Scale SVM Learning Practical. In B. Scholkopf, C. J. C. Burges & A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, USA, 1998. **http://www.kernel-machines.org**
15. Johnson, R.A. and Wichern, D.W. (1992). Applied Multivariate Statistical Analysis. Prentice-Hall International, Inc, Third Edition.
16. Kornienko, L. and Dowe, D.L. and Albrecht, D.W. (2002). Message Length Formulation of Support Vector Machines for Binary Classification - A Preliminary Scheme in Lecture Notes in Artificial Intelligence (LNAI) 2557, 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia. Springer-Verlag.
17. Kornienko, L. (2005). Implementing a Support Vector Machine in a Message Length Framework. Masters Thesis, School of Information Technology, Monash University, Clayton, Australia.
18. Kullback, S. (1959) *Information Theory and Statistics*. John Riley and Sons, Inc.
19. Mangasarian, O. L. and Musicant, D. R. (2000). *Lagrangian Support Vector Machines*. (Technical Report 00-06). Data Mining Institute. **http://www.kernel-machines.org**
20. Morrison, D.F. (1990). Multivariate Statistical Methods. McGraw-Hill International Editions, Third Edition.
21. Needham, S.L. and Dowe, D.L. (2001). Message Length as an Effective Ockham's Razor in Decision Tree Induction. Proc. 8th International Workshop on Artificial Intelligence and Statistics (AI+STATS 2001), pp 253-260, Key West, Florida, U.S.A., Jan. 2001.
22. Platt, J. (1999). *Sequential minimal optimization: A fast algorithm for training support vector machines* in *Advances in Kernel Methods - Support Vector Learning*, Bernhard Scholkopf, Christopher J. C. Burges and Alexander J. Smola, Eds. 1999, pp. 185-208, MIT Press.

23. Rissanen, J. J. (1978). Modeling by shortest data description. *Automatica*, 14, 465–471.
24. Tan, P.J. and Dowe, D.L. (2002). MML Inference of Decision Graphs with Multi-Way Joins, in Lecture Notes in Artificial Intelligence (LNAI) 2557, 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia. Springer-Verlag.
25. Tan, P.J. and Dowe, D.L. (2003). MML Inference of Decision Graphs with Multi-Way Joins and Dynamic Attributes, in Lecture Notes in Artificial Intelligence, T.D.Gedeon and L.C.C.Fung (eds.), 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia. pp. 269-281, Springer-Verlag. ISSN:0302-9743, ISBN:3-540-20646-9, Vol 2903.
26. Tan, P.J. and Dowe, D.L. (2004). MML Inference of Oblique Decision Trees in Lecture Notes in Artificial Intelligence, G.I.Webb and X.Yu, Eds., 17th Australian Joint Conference on Advances in Artificial Intelligence, Cairns, Australia. pp. 1082-1088, Springer-Verlag. ISSN:0302-9743, ISBN:3-540-24059-4, Vol 3339.
27. Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, New York.
28. Viswanathan, M. and Wallace, C.S. (1999). A Note on the Comparison of Polynomial Selection Methods, in Proceedings of Uncertainty 99: The Seventh International Workshop on Artificial Intelligence and Statistics, D. Heckerman and J. Whittaker (eds.), pp. 169-177, Morgan Kaufmann Publishers, Inc.,San Francisco, CA, USA, Fort Lauderdale, Florida, January 1999.
29. Viswanathan, M. and Wallace, C.S. and Dowe, D.L. and Korb, K.B. (1999). Finding Cutpoints in Noisy Binary Sequences - A Revised Empirical Evaluation, pp 405-416, Proc. 12th Australian Joint Conf. on Artif. Intelligence, Lecture Notes in Artificial Intelligence (LNAI) 1747, Sydney, Australia, December 1999.
30. Wallace, C.S. (1995). MML inference of predictive trees, graphs and nets, Applied Decision Technologies, Proceedings of applied decision technologies (ADT) 1995 conference, Stream 1: Computational learning and probabilistic reasoning London, pp 101-15, 1995.
31. Wallace, C.S. (2005). Statistical and Inductive Inference by Minimum Message Length, Springer. ISBN: 0-387-23795-X
32. Wallace, C. S. and Boulton, D.M. (1968). An information measure for classification. *Computer Journal*, 11, 185-194.
33. Wallace, C.S. and Dowe, D.L. (1993). MML Estimation of the von Mises Concentration Paramter, Computer Science Department, Monash University, 93/193.
34. Wallace, C.S. and Dowe, D.L. (1994). Intrinsic classification by MML - the Snob program. Proc. 7th Australian Joint Conf. on Artificial Intelligence, UNE, Armidale, Australia, November, pp37-44.
35. Wallace, C .S. and Dowe, D. L. (1999). Minimum Message Length and Kolmogorov Complexity. *Computer Journal Special Issue: Kolmogorov Complexity*. 42(4), pp. 270-283.
36. Wallace, C.S. and Dowe, D.L. (1999). Refinements of MDL and MML coding. *Computer Journal Special Issue: Kolmogorov Complexity*. C J van Rijsbergen (ed). 42(4), pp. 330-337, ISSN: 0010-4620.
37. Wallace, C.S. and Dowe, D.L. (1999). Rejoinder. *Computer Journal Special Issue: Kolmogorov Complexity*. C J van Rijsbergen (ed). 42(4), pp. 345-347, ISSN: 0010-4620.
38. Wallace, C.S. and Dowe, D.L. (2000). MML Clustering of Multi-State, Poisson, von Mises Circular and Gaussian Distributions. *Statistics and Computing*, 10(1), pp. 73-83.
39. Wallace, C. S. and Freeman, P.R. (1987). Estimation and Inference by Compact Coding, *J Royal Stat. Soc. B*. 49, 240-252.

40. Wallace, C.S. and Freeman, P.R. (1992). Single factor analysis by MML estimation, J Royal Stat. Soc. B. 54, 1, 195-209, 1992.
41. Wallace, C.S. and Patrick, J.D. (1993). Coding Decision Trees. *Machine Learning*, 11, 7-22.