Proc. IEEE Conference on System, Man and Cybernetics, Washington DC, October 2003, pp. 1400-1405

Use of Machine Learning for Classification of Magnetocardiograms*

Mark Embrechts, Boleslaw Szymanski
Center for Pervasive Computing and Networking
Rensselaer Polytechnic Institute
Troy, NY, U.S.A.
{embrem,szymab}@rpi.edu

Karsten Sternickel, Thanakorn Naenna, Ramathilagam Bragaspathi
Cardiomag Imaging Inc.
Schenectady, NY, U.S.A.
karsten@cardiomag.com

Abstract - *We describe the use of machine learning for pattern recognition in magnetocardiography (MCG) that measures magnetic fields emitted by the electrophysiological activity of the heart. We used direct kernel methods to separate abnormal MCG heart patterns from normal ones. For unsupervised learning, we introduced Direct Kernel based Self-Organizing Maps. For supervised learning we used Direct Kernel Partial Least Squares and (Direct) Kernel Ridge Regression. These results are then compared with classical Support Vector Machines and Kernel Partial Least Squares. The hyper-parameters for these methods were tuned on a validation subset of the training data before testing. We also investigated the most effective pre-processing, using local, vertical, horizontal and two-dimensional (global) Mahanalobis scaling, wavelet transforms and experimented with variable selection by filtering. The results, similar for all three methods, were encouraging, exceeding the quality of classification achieved by the trained experts.*

Keywords: Classification, pre-processing, direct kernel methods, support vector machines, self-organized maps.

1 Introduction

In this paper, we describe the use of direct-kernel methods and support vector machines for pattern recognition in magnetocardiography (MCG) that measures magnetic fields emitted by the electrophysiological activity of the human heart. A SQUID-based measuring device for MCG that can be used in regular, magnetically unshielded hospital rooms is currently under development. The operation of the system is computer-controlled and largely automated. Proprietary software is used for precise 24-bit control and data acquisition followed by filtering, averaging, electric/magnetic activity localization, heart current reconstruction, and derivation of diagnostic scores**.**

The interpretation of MCG recordings remains a challenge since there are no databases available from which precise rules could be educed. Hence, we studied the methods to automate interpretation of MCG measurements to minimize human input for the analysis. In this paper, we report our results on detecting ischemia, a condition arising in many common heart diseases that may result in heart attack, the leading cause of death in the United States.

The paper is organized as follows. We start with a discussion of data acquisition and preprocessing in the next section. We discuss what kind of preprocessing is suitable to different learning methods. Section 3 presents the core of our results: the comparison of performance of different machine learning techniques for our problem, and methodologies for assessment of prediction quality and for the regularization parameter selection. Section 4 discusses feature selection. The final section provides conclusions and outlines the future work in this area.

2 Data acquisition and pre-processing

MCG data are acquired at 36 locations above the torso by making four sequential measurements in mutually adjacent positions. In each position the nine sensors measure the cardiac magnetic field for 90 seconds using a sampling rate of 1000 Hz leading to 36 individual time series. For diagnosis of ischemia, a bandwidth of 0.5 Hz to 20 Hz is needed, so a hardware low pass filter at 100 Hz using 6th-order Bessel filter characteristics is applied, followed by an additional digital low pass filter at 20 Hz using the same characteristics, but a higher order. To eliminate remaining stochastic noise components, the complete time series is averaged using the maximum of the R peak of the cardiac cycle as a trigger point. For automatic classification, we used data from a time window between the J point and T peak [5] of the cardiac cycle in which values for 32 evenly spaced points were interpolated from the measured data. The training data consist of 73 cases that were easy to classify visually by trained experts. The testing was done on a set of 36 cases that included patients whose magnetocardiograms misled or confused trained experts doing visual classification.

Data were preprocessed in this case by first subtracting the bias from each signal. Then, we investigated the most effective pre-processing for our multi-variate time-series signals, including local, vertical, horizontal and two-

dimensional (global) Mahanalobis scaling, and wavelet transforms. An important consideration was preservation of data locality, which was achieved by applying the Daubechies-4 wavelet transform to each signal [3]. It was chosen, because of the relatively small set of data (32) in each of the interpolated time signals. Only SOM and K-PLS methods that observe data locality in input did not require this transformation. Next, we Mahalanobis scaled the data, first on all 36 signals and then (for all except SOM based methods) vertically. A typical dataset for 36 signals that are interpolated to 32 equally spaced points in the ST segment [5] and after Mahalanobis scaling on each of the individual signals is shown in Fig. 1.

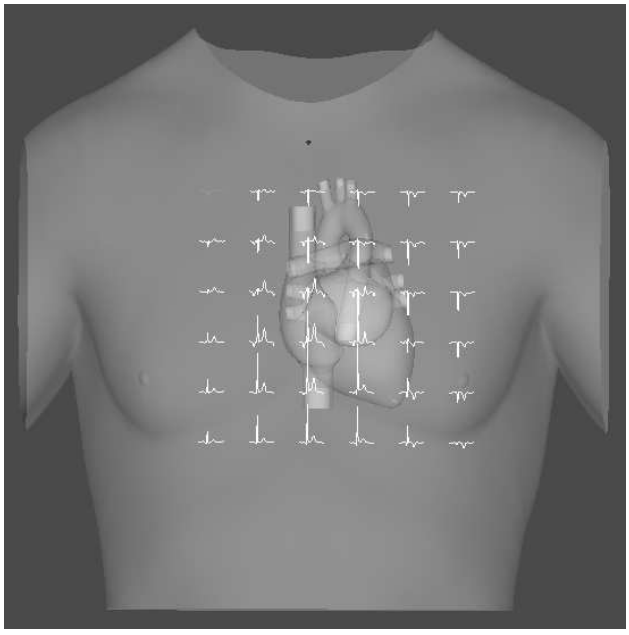

Figure 1. Filtered and averaged temporal MCG traces over one cardiac cycle collected in 36 channels (the 6x6 grid).

3 Predicitive modeling for MCG data classification

We investigated both unsupervised and supervised learning methods. For the former, we used Direct Kernel (DK)-SOMs, since SOMs are often applied for novelty detection and automated clustering. Our DK-SOM has a 9×18 hexagonal grid with unwrapped edges. For supervised learning, we used four kernel-based regression algorithms: classical Support Vector Machines effective in extracting relevant parameters from complex data spaces, kernel partial least square K-PLS, as proposed by Rosipal [10], direct kernel partial least square (DK-PLS), and Least-Squares Support Vector Machines (i.e., LS-SVM, also known as kernel ridge regression).

Support Vector Machines or SVMs have proven to be formidable machine learning tools because of their efficiency, model flexibility, predictive power, and theoretical transparency [2,11,15]. While the nonlinear properties of SVMs can be exclusively attributed to the kernel transformation, other methods, such as self-organizing maps or SOMs [9], are inherently nonlinear because they incorporate various neighborhood-based manipulations. Unlike SVMs, the prime use for SOMs is often as a visualization tool [4] for revealing the underlying similarity/cluster structure of high-dimensional data on a two-dimensional map, rather than for regression or classification predictions.

We used the Analyze/StripMiner software package, developed in-house for the analysis [14], but made use of SVMLib [1] for the SVM model. Using the training set, we optimized the values for the parameters in DK-SOM, SVM, DK-PLS and LS-SVM before testing. The results are similar to the quality of classification achieved by the trained experts and similar for all tested methods, even though they use different data preprocessing. This is important because it indicates that there was no over-training in any of the tested methods. The agreement between DK-PLS, SVMLib, and LS-SVM is particularly good, and there are no noticeable differences between these methods on these data. The results are shown in Tables 1-2. Table 1 lists the number of correctly classified patterns and the number of misses for the negative and positive cases. Table 2 provides additional mesaures of quality of prediction.

Table 1. Numbers of correct patterns and misses (for negative and positive cases on 36 test data) as well as execution times for magnetocardiogram data. SVMLib and K-PLS used time domain and the remaining methods used D-4 wavlet domain.

Method	%correct	#misses	time (s)
SVMLib	74	4+5	10
K-PLS	74	4+5	6
DK-PCA	71	7+3	5
PLS	63	2+11	3
K-PLS	80	2+5	6
DK-PLS	**83**	**1+5**	5
SVMLib	80	2+5	10
LS-SVM	80	2+5	**0.5**
SOM	63	3+10	960
DK-SOM	71	5+5	28
DK-SOM	77	3+5	28

After tuning, σ for SVM was chosen as 10. The regularization parameter, C, in SVMLib was set to $1/\lambda$ as suggested in [10]. Based on our experience with other applications [14] and scaling experiments, the value of λ was determined from the following equation:

$$\lambda = \min\left\{1;\ \left(\frac{n}{1500}\right)^{\frac{3}{2}}\right\} \qquad (1)$$

The agreement between the direct kernel methods (DK-PLS and LS-SVM), K-PLS, and the traditional kernel-based SVM (SVMLib) indicates a near-optimal choice for the ridge parameter resulting from this formula.

3.1 Metrics for Assessing the Model Quality

For a regression problem, another way to capture the error is by the *Root Mean Square Error* index or *RMSE*, which is defined as the average value of the squared error (either for the training set or the test set) according to:

$$RMSE = \sqrt{\frac{1}{n}\sum_i \left(\hat{y}_i - y_i\right)^2} \qquad (2)$$

While the root mean square error is an efficient way to compare the performance of different prediction methods on the same data, it is not an absolute metric in the sense that the *RMSE* will depend on how the response for the data was scaled. In order to overcome this handicap, we also used additional error measures that are less dependent on the scaling and magnitude of the response value. A first metric that we used for assessing the quality of a trained model is r^2, which is defined as the squared coefficient of correlation between target values and predictions for the response according to:

$$r^2 = \frac{\sum_{i=1}^{n_{\text{train}}} \left(\hat{y}_i - \bar{y}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{n_{\text{train}}} \left(\hat{y}_i - \bar{y}\right)^2}\sqrt{\sum_{i=1}^{n_{\text{train}}} \left(y_i - \bar{y}\right)^2}} \qquad (3)$$

where n_{train} represents the number of data points in the training set. r^2 takes values between zero and unity, and the higher the r^2 value, the better the model. An obvious drawback of using r^2 for assessing the model quality is that it only expresses a linear correlation, indicating how well the predictions follow a line if $\hat{y}$ is plotted as function of y. While one would expect a nearly perfect model when r^2 is unity, this is not always the case. A second, and more powerful measure to assess the quality of a trained model is the so-called "Press *r* squared", or R^2, often used in chemometric modeling [6], where R^2 is defined as [7]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{train}} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n_{train}} \left(y_i - \bar{y}\right)^2} \qquad (4)$$

We consider R^2 a better measure than r^2, because it accounts for the residual error as well. Just like r^2, R^2 ranges between zero and unity, and higher the value for R^2, better the model. The R^2 metric is generally smaller than r^2. For large datasets, R^2 tends to converge to r^2, and the comparison between r^2 and R^2 for such data often reveals hidden biases.

Table 2. Quality measures for different methods for creating predictive models for magneto-cardiogram data.

Method	q2	Q2	RMSE
SVMLib	0.767	0.842	0.852
K-PLS	0.779	0.849	0.856
DK-PCA	0.783	0.812	0.87
PLS	0.841	0.142	1.146
K-PLS	0.591	0.694	0.773
DK-PLS	**0.554**	**0.662**	**0.75**
SVMLib	0.591	0.697	0.775
LS-SVM	0.59	0.692	0.772
SOM	0.866	1.304	1.06
DK-SOM	0.855	1.0113	0.934
DK-SOM	0.755	0.859	0.861

For assessing the quality of the validation set or a test set, we introduce similar metrics, q^2 and Q^2, where q^2 and Q^2 are defined as $1-r^2$ and $1-R^2$, respectively, for the data in the test set. For a model that perfectly predicts on the test data we would expect q^2 and Q^2 to be zero. The reason for introducing metrics that are symmetric between the training set and the test set is actually to avoid confusion. Q^2 and q^2 values apply to a validation set or a test set, and we would expect these values to be quite low in order to have a good predictive model. R^2 and r^2 values apply to training data, and it is easy to notice that if the predictions are close to actual values, they both are close to unity. Hence, any of them significantly different from 1 indicates a model with poor predictive ability.

Linear methods, such as partial-least squares, result in inferior predictive models as compared to the kernel methods. For K-PLS and DK-PLS, we chose 5 latent variables, but the results were not critically dependent on the exact choice of the number of latent variables. We also tried Direct Kernel Principal Component Analysis (DK-PCA), the direct kernel version of K-PCA [11-12,16], but the results were more sensitive to the choice of the number of principal components and not as good as the ones obtained using other direct kernel methods.

Typical prediction results for the magnetocardiogram data based on wavelet transformed data and DK-PLS are shown in Fig. 2. We can see from this figure that six data

points are misclassified altogether in the predictions (one healthy or negative case and five ischemia cases). These cases were also difficult to identify correctly for the trained expert, based on a 2-D visual display of the time-varying magnetic field, obtained by proprietary methods.

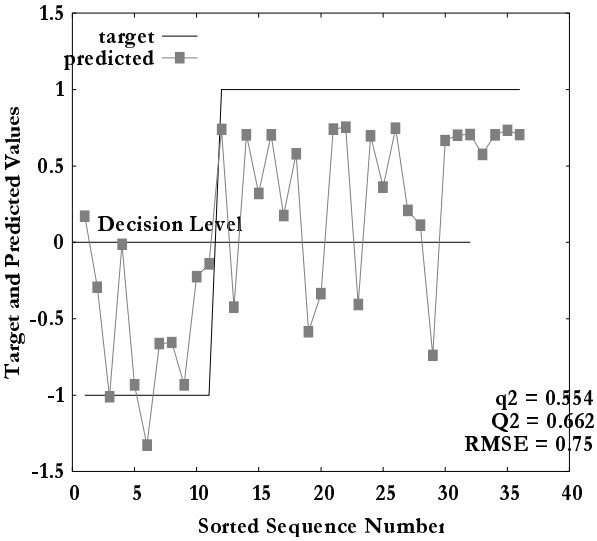

Figure 2. Error plot for 35 test cases, based on K-PLS for wavelet-transformed data.

For medical data, it is often important to be able to make a trade-off between false negative and false-positive cases, or between sensitivity and specificity (which are different metrics related to false positives and false negatives). In machine-learning methods, such a trade-off can easily be accomplished by changing the threshold for interpreting the classification. For example, in Fig. 2, rather than using zero as the discrimination value, one could shift the discrimination threshold towards a more desirable level, hereby influencing the false positive/false negative ratio.

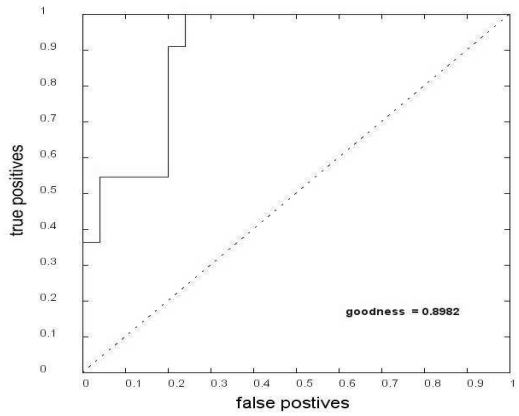

Figure 3. ROC curve showing possible trade-offs between false positive and false negatives.

A summary of all possible outcomes of such changes in the discrimination value can be displayed in an ROC curve, as shown in Fig. 3 for the above case. The concept of ROC curves (or Receiver Operator Characteristics) originated from the early development of the radar in the 1940's for identifying airplanes and is summarized in [13].

Figure 4 displays a projection of 73 training data, based on (a) Direct Kernel Principal Component Analysis (DK-PCA), and (b) Direct Kernel PLS (DK-PLS). Diseased cases are shown as filled circles. Figure 4b shows a clearer separation and wider margin between different classes, based on the first two components for DK-PLS as compared to results of DK-PCA that are shown in Figure 4a. The test data, originally shown on these pharmaplots as dark and light crosses, shows an excellent separation between healthy and diseased cases for both methods.

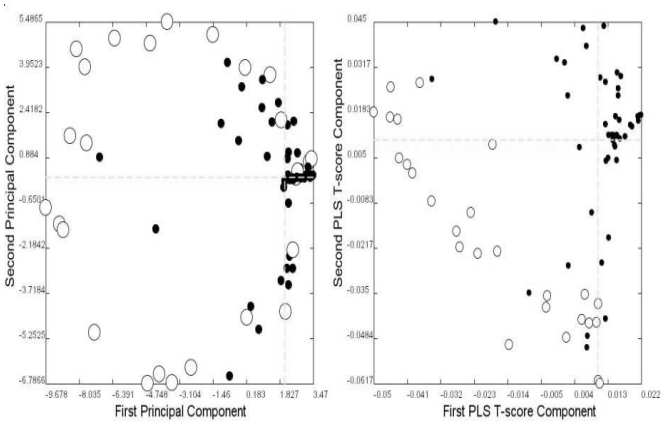

Figure 4. Projection of 73 training data, based on (a) Direct Kernel Principal Component Analysis (DK-PCA), and (b) Direct Kernel PLS (DK-PLS). Diseased cases are shown as filled circles (the test data are not shown).

A typical 9×18 self-organizing map on a hexagonal grid in wrap-around mode, based on the direct kernel SOM, is shown in Figure 5. The wrap-around mode means that the left and right boundaries (and also the top and bottom boundaries) flow into each other, and that the map is an unfurling of a toroidal projection. The dark hexagonals indicate diseased cases, while the light hexagonals indicate healthy cases. Fully colored hexagons indicate the positions for the training data, while the white and dark-shaded numbers are the pattern identifiers for healthy and diseased test cases. Most misclassifications actually occur on boundary regions in the map. The cells in the map are colored by semi-supervised learning, i.e., each data vector, containing 36x32 or 1152 features, is augmented by an additional field that indicates the color. The color entry in the data vectors are updated in a similar way as for the weight vectors, but they are not used to calculate the distance metrics for determining the winning cell. The resulting maps for a regular SOM implementation are very similar to those obtained with the direct kernel DK-SOM.

The execution time for generating DK-SOM on a 128 MHz Pentium III computer was 28 seconds, rather than 960 seconds required for generating the regular SOM, because the data dimensionality dropped to effectively 73 (the number of training data) from the original 1152, after the kernel transformation on the data. The fine-tuning for the SOM and DK-SOM was done in a supervised mode with learning vector quantization [9]. While the results based on SOM and DK-SOM are still excellent, they are not as good as those obtained with the other kernel-based methods (SVMLib, LS-SVM, and K-PLS).

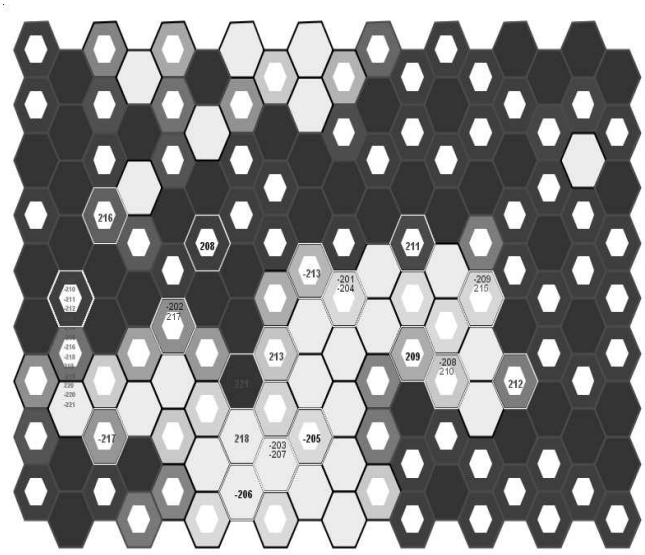

Figure 5. Test data displayed on a self-organizing map based on a 9×18 DK-SOM in wrap-around mode. The dark hexagonals indicate diseased cases, while the light hexagonals indicate healthy cases. Fully colored hexagons indicate the positions for the training data, while the red and blue numbers are the pattern identifiers for healthy and diseased test cases. Negative numbers correspond to pre-stress and positive number to post-stress measurements for each patient.

4 Feature selection

The results presented in the previous section were obtained using all 1152 (36×32) descriptors. It would be most informative to the domain expert, if we were able to identify where exactly in the time or wavelet signals and for which of the 36 magnetocardiogram signals that were measured at different positions for each patient the most important information necessary for good binary classification is located. Such information can be derived by feature selection.

Feature selection, i.e., the identification of the most important input parameters for the data vector, can proceed in two different ways: the filtering mode and the wrap-around mode. In the filtering mode, features are eliminated based on a prescribed, and generally unsupervised procedure. An example of such a procedure could be the elimination of descriptor columns that contain four σ outliers, as is often the case in PLS applications for chemometrics. It is also common to drop "cousin" descriptors in a filtering mode, i.e., features that show more than 95% correlation with another descriptor. Depending on the modeling method, it is often common practice to drop the cousin descriptors and only retain the descriptors that (i) either show the highest correlation with the response variable, or (ii) have the clearest domain transparency to the domain expert for explaining the model.

The second mode of feature selection is based on the wrap-around mode. One wants to retain only the most relevant features necessary to have a good predictive model. Often, the modeling quality improves after the proper selection of the optimal feature subset. Determining the right subset of features can proceed based on different concepts, and the resulting subset of features is often dependent on the modeling method. Feature selection in a wrap-around mode generally proceeds by using a training set and a validation set, and the validation set is used to confirm that the model is not over-trained by selecting a spurious set of descriptors. Two generally applicable methods for feature selections are based on the use of genetic algorithms and sensitivity analysis.

The idea with the genetic algorithm approach is to be able to obtain an optimal subset of features from the training set, showing a good performance on the validation set as well.

The concept of sensitivity analysis [8] exploits the saliency of features, i.e., once a predictive model has been built, the model is used for the average value of each descriptor, and the descriptors are tweaked, one-at-a time between a minimum and maximum value. The sensitivity for a descriptor is the change in predicted response. The premise is that when the sensitivity for a descriptor is low, it is probably not an essential descriptor for making a good model. A few of the least sensitive features can be dropped during one iteration step, and the procedure of sensitivity analysis is repeated many times until a near optimal set of features is retained. Both the genetic algorithm approach and the sensitivity analysis approach are true soft computing methods and require quite a few heuristics and experience. The advantage of both approaches is that the genetic algorithm and sensitivity approach are general methods that do not depend on the specific modeling method.

5 Conclusions

The binary classification of MCG data represents a challenging problem for various reasons: the quantity of the data is low, the quality of the data varies from hospital to

hospital, and the patient classification by the "gold standard" is not 100% correct. Applying standard machine learning techniques such as SOM and SVM already exceeds the predictive accuracy of a standard ECG in these cases (74% vs. 50%). The break-through, though, was achieved by first transforming the data into the wavelet domain, and then, additionally, applying a kernel transformation to wavelet coefficients (this increased the predictive accuracy to 83% vs. 74% achieved by the standard methods and the original 50% achieved by ECG).

The agreement of the results between kernel PLS (K-PLS) as proposed by Rosipal [10], direct kernel PLS (DK-PLS), support vector machine (SVMLib), and least square SVM (LS-SVM) is generally excellent. In this case, DK-PLS gave a superior performance, but the differences between kernel-based methods are not significant. This excellent agreement shows the robustness of the direct kernel methods. It could only be achieved if the selection of the ridge parameter by Eq. (1) was nearly optimal. This selection defines also the regularization parameter, C, in support vector machines, when C is taken as $1/\lambda$.

The obtained results are meaningful for medical community. With DK-PLS we reached a sensitivity of 92% and a specificity of 75% for the detection of ischemia defined by coronary angiography. It is of notice that MCG is a purely *functional* tool that is sensitive for abnormalities in the electrophysiology of the heart and, therefore, can only diagnose the *effect* of a disease. The gold standard (coronary angiography), however, is a purely *anatomical* tool and diagnoses the *cause* of ischemic heart disease. Since MCG detects abnormalities that are not visible to the gold standard, it will always produce "false positives", which explains the comparatively low specificity in this application.

The continuation of this work will include three directions. First, we will develop feature selection techniques based on large number of patients. For ischemia, we will attempt to recognize where the source of blockage causing ischemia is located in the body of a patient. Finally, we will also investigate a multi-class problem in which different heart diseases, not just ischemia, will be classified by our machine learning methods. Such multi-class problems are particularly challenging for SVM based method when the classes are non-ordinal, as is the case for heart diseases.

6 Acknowledgement

The authors acknowledge the National Science Foundation support of this work (IIS-9979860) and SBIR Phase I #0232215. The content of this paper does not necessarily reflect the position or policy of the U.S. Government or Cardiomag Imaging Inc. - no official endorsement should be inferred or implied.

References

[1] C.-C. Chang and C.-J. Lin, LibSVM, OSU, see http://www.csie.ntu.edu.tw/~cjlin/libsvmSVMLib.

[2] N. Cristianini and J. Shawe-Taylor [2000] *Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press.

[3] I. Daubechies [1992], *Ten Lectures on Wavelets*, Siam, Philadelphia, PA.

[4] G. Deboeck and T. Kohonen (Eds.) [1998] *Visual Explorations in Finance with Self-Organizing Maps*, Springer.

[5] V. Froelicher, K. Shetler, and E. Ashley [2002] "Better Decisions through Science: Exercise Testing Scores." *Progress in Cardiovascular Diseases*, Vol. 44(5), pp. 385-414.

[6] A. Golbraikh and A. Tropsha [2002] "Beware of q^2!" *Journal of Molecular Graphics and Modelling*, Vol 20, pp. 269-276.

[7] R. A. Johnson and D. W. Wichern [2000] *Applied Multivariate Statistical Analysis, 2 ed.*, Prentice Hall.

[8] R. H. Kewley, and M. J. Embrechts [2000] "Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks," *IEEE Transactions on Neural Networks*, Vol.11 (3), pp. 668-679.

[9] T. Kohonen [1997] *Self-Organizing Maps, 2nd Edition*, Springer.

[10] R. Rosipal and L. J. Trejo [2001] "Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Spaces," *Journal of Machine Learning Research*, Vol. 2, pp. 97-123.

[11] B. Schölkopf and A. J. Smola [2002] *Learning with Kernels*, MIT Press.

[12] B. Schölkopf, A. Smola, and K-R Müller [1998] "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, Vol. 10, 1299-1319, 1998.

[13] J. A. Swets, R. M. Dawes, and J. Monahan [2000] "Better Decisions through Science," *Scientific American*, pp. 82-87.

[14] The Analyze/StripMiner, the description and the code are available at http://www.drugmining.com.

[15] V. Vapnik [1998] *Statistical Learning Theory*, John Wiley & Sons.

[16] W. Wu, D. L. Massarat and S. de Jong [1997] "The Kernel PCA Algorithm for Wide Data. Part II: Fast Cross-Validation and Application in Classification of NIR Data," *Chemometrics and Intelligent Laboratory Systems*, Vol. 37, pp. 271-280.