Variable Star Classification Using Multi-View Metric Learning

K.B. Johnston,[1] S.M. Caballero-Nieves,[1] A.M. Peter,[2] V. Petit, [3] and R. Haber[4]
[1]*Aerospace, Physics and Space Sciences Dept., Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL, US;* kyjohnst2000@my.fit.edu

[2]*Computer Engineering and Sciences Dept., Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL, US*

[3]*Physics and Astronomy Dept., University of Delaware, Newark, DE, USA*

[4]*Mathematical Sciences Dept., Florida Institute of Technology, 150 W. University Blvd., Melbourne, FL, US*

Abstract. Comprehensive observations of variable stars can include time domain photometry in a multitude of filters, spectroscopy, estimates of color (e.g. U-B), etc. When it is considered that the time domain data can be further transformed via digital signal processing methodologies, the potential representations of the observed target star are limitless. Presented here is an initial review of multi-view classification as applied to variable star classification, to address this challenge.

1. Introduction

The classification of variable stars relies on a proper selection of features of interest and a classification framework that can support the linear separation of those features. Features should be selected that quantify the signature of the variability, i.e. its' structure and information content. Prior studies have generated a multitude of features (e.g., SSMM, Fourier Transform, Wavelet Transformation, DF, etc.) that attempt to completely differentiate or linearly separate various variable stars class types (Richards et al. 2012; Graham et al. 2013; Mahabal et al. 2017; Hinners et al. 2018). How to process the complete set of features is an outstanding question.

Metric Learning has a number of benefits that are advantageous to the astronomer. First, metric learning uses k-NN classification to generate the decision space, k-NN provides instant clarity into the reasoning behind the classifiers decision (based on similarity, "x_i is closer to x_j than x_k"). Second, metric learning leverages side information (the supervised labels of the training data) to improve the metric, i.e. a transformation of the distance between points that favors the proposed goals: pull representatives from similar classes closer together and push representatives from different classes further apart, optimize to a low complexity metric via regularization, allow for feature dimensionality reduction, etc. (Bellet et al. 2015). Third, k-NN implemented as part of metric learning can be supported by other structures such as partitioning methods to allow for a rapid response time, despite a high number of training data (Faloutsos et al. 1994). Lastly, it can support the development of an anomaly detection functionality, which

has been shown to be necessary to generate meaningful data in astronomical datasets (Johnston & Peter 2017).

Multi-view learning can be leveraged to address the multitude of feature spaces or views that may be available to the astronomer for the purpose of classification. Multi-view learning can be roughly divided into three topic areas: 1) co-training, 2) multiple-kernel learning, and 3) subspace learning. This work will focus on the method of co-training, specifically metric co-training. The multi-view metric distance is defined as Equation 1:

$$d_M^2(x_i, x_j) = \sum_{k=1}^{K} w_k \left(x_i^k - x_j^k\right)^T \mathbf{M}_k \left(x_i^k - x_j^k\right) \tag{1}$$

where K is the number of views, x_i^k is the i^{th} observation and the k^{th} view for a given input. Presented here is a design the incorporates both metric learning and multi-view learning.

2. Theory and Design

Our proposal is an implementation of both the feature extraction and classifier for the purposes of multi-class identification, that can handle raw observed data. We implement two novel time domain feature space transforms, SSMM (Johnston & Peter 2017) and DF (Helfer et al. 2015), to demonstrate the utility of the metric learning and multi-view learning. It is not suggested that these features are going to be the best in all cases, nor are they the only choice as is apparent from Fulcher et al. (2013).

Large Margin Multi-Metric Learning (Hu et al. 2014, 2017) is an example of metric co-training; the designed objective function minimizes the objective function of the individual view, as well as the difference between view distances, simultaneously. The objective function for LM^3L is defined as Equation 2:

$$\begin{gathered} \min_{\mathbf{M}_1,\ldots,\mathbf{M}_K} J = \sum_{k=1}^{K} w_k^p I_k + \lambda \sum_{k,l=1,k<l}^{K} \sum_{i,j} \left(d_{\mathbf{M}_k}^2(x_i^k, x_j^k) - d_{\mathbf{M}_l}^2(x_i^l, x_j^l)\right)^2 \\ s.t. \quad \sum_{k=1}^{K} w_k = 1, w_k \geq 0, \lambda > 0 \end{gathered} \tag{2}$$

where I_k is the objective function for a given k^{th} individual view (Equation 3):

$$\min_{\mathbf{M}_k} I_k = \sum_{i,j,} h\left(\tau_k - y_{ij}\left(\mu_k - d_{\mathbf{M}_k}^2(x_i^k, x_j^k)\right)\right) \tag{3}$$

where $h(x) = \max(x, 0)$ is the hinge loss function, $y_{ij} = 1$ when data are from the same class and $y_{ij} = -1$ otherwise, and τ_k and μ_k are threshold parameters that enforce the constraint $y_{ij}\left(\mu_k - d_{\mathbf{M}_k}^2(x_i^k, x_j^k)\right) > \tau_k$. In practice, optimizing $\mathbf{M}_k$ requires enforcing the requirement $\mathbf{M}_k \succ 0$, which can be slow depending on the methodology used. Hu et al. (2014) transform the metric $\mathbf{M_k}$, following Weinberger et al. (2006), as $\mathbf{M} = \mathbf{L}^T\mathbf{L}$.

The algorithm operates as a two step process (alternating optimization) between the optimization of the decomposed metrics $\mathbf{L}_k$ and the weighting between the views w_k. The iterative update to the $\mathbf{L}_k$ estimate is generated via gradient for each view. Second, the metrics $\mathbf{M}_k$ are fixed with the updated values and the individual weights

$w = [w_1, w_2, ...w_k]$ are estimated. The estimates for each weight can be given as Equation 4:

$$w_k = \frac{(1/I_k)^{1/(p-1)}}{\sum_{k=1}^{K} (1/I_k)^{1/(p-1)}} \tag{4}$$

These two steps are then repeated for each iteration until $\left|J^{(t)} - J^{(t-1)}\right| < \varepsilon$, i.e. some minimum is reached. The derivation of this algorithm is outlined in Hu et al. (2014), and the algorithm for optimization for LM^3L is given as their Algorithm 1..

2.1. Large Margin Multi-Metric Learning with Matrix Variates ($LM^3L - MV$)

Glanz & Carvalho (2013) define the matrix normal distribution as $X_i \sim MN(\mu, \mathbf{\Sigma}_s, \mathbf{\Sigma}_c)$, where X_i and μ are $p \times q$ matrices, $\mathbf{\Sigma}_s$ is a $p \times p$ matrix defining the row covariance, and $\mathbf{\Sigma}_c$ is a $q \times q$ matrix defining the column covariance. The Mahalanobis distance for the Matrix-Variate Multi-View case is given as Equation 5:

$$d_{\mathbf{U}_k,\mathbf{V}_k}(X_i^k, X_j^k) = \mathrm{tr}\left[\mathbf{U}_k \left(X_i^k - X_j^k\right)^T \mathbf{V}_k \left(X_i^k - X_j^k\right)\right] \tag{5}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ represents the covariance of the column and row respectively. The individual view objective function is constructed similar to the LMNN Weinberger et al. (2006) methodology; the joint, sub-view objective function is then Equation 6:

$$\min_{\mathbf{U}_k,\mathbf{V}_k} I_k = \sum_{i,j} \eta_{ij}^k \cdot d_{\mathbf{U}_k,\mathbf{V}_\mathbf{k}}(X_i^k, X_j^k) + \gamma \sum_{j \rightsquigarrow i,l} \eta_{ij}^k (1 - y_{il}) \cdot h\left[d_{\mathbf{U}_k,\mathbf{V}_k}(X_i^k, X_j^k) - d_{\mathbf{U}_k,\mathbf{V}_k}(X_i^k, X_l^k) + 1\right] + \frac{\lambda}{2} \|\mathbf{U}_k\|_F^2 + \frac{\lambda}{2} \|\mathbf{V}_k\|_F^2 \tag{6}$$

Similar to LM^3L the objective function is Equation 7:

$$\min_{\mathbf{U}_k,\mathbf{V}_k} J_k = w_k I_k + \mu \sum_{q=1, q \neq k}^{K} \sum_{i.j} \left(d_{\mathbf{U}_k,\mathbf{V}_k}(X_i^k, X_j^k) - d_{\mathbf{U}_l,\mathbf{V}_l}(X_i^q, X_j^q)\right)^2 \tag{7}$$

This objective design can be solved using gradient descent solver; to enforce the requirements of $\mathbf{U}_k \succ 0$ and $\mathbf{V}_k \succ 0$ we leverage the decomposition $\mathbf{U}_k = \mathbf{\Gamma}_k^T \mathbf{\Gamma}_k$ and $\mathbf{V}_k = \mathbf{N}_k^T \mathbf{N}_k$ and find the gradient of the objective function with respect to the decomposed matrices $\mathbf{\Gamma}_k$ and $\mathbf{N}_k$. Weights per view can be estimated using the same procedure as in LM^3L. The implementation of distance in the multi-view case, i.e. implementation of distance used in the k-NN algorithm is just the weighted average of Equation 5 over all views.

3. Conclusion

Optimal parameters are found for the LM^3L algorithm LINEAR data (following a standard 5-fold cross-valdiation procedure). The trained classifier is applied to the test data, the confusion matrices resulting from the application to LINEAR data are presented as an example in Table 1:

The classification of variable stars relies on a proper selection of features of interest and a classification framework that can support the linear separation of those features.

Table 1. LINEAR Confusion Matrix via LM^3L

Error Rate	RL (ab)	δ S / SP	Al	RL (c)	CB	Miss
RR Lyr (ab)	0.9927	0	0	0.00648	0.0009	0
δ Scu / SX Phe	0.0370	0.9259	0	0	0	0.037
Algol	0.0073	0	0.7737	0	0.218	0
RR Lyr (c)	0.0485	0	0.0027	0.9434	0.0054	0
Contact Binary	0.0034	0	0.0377	0.0011	0.9577	0

Features should be selected that quantify the signature of the variability, i.e. its' structure and information content. To support the set of high-dimensionality features, or views, multi-view metric learning is investigated as a viable design. Multi-view learning provides an avenue for integrating multiple transforms to generate a superior classifier. Future research will include methods for addressing high dimensionality matrix data (e.g. SSMM), applying the designed classifier (LM^3L) to the datasets, improving the parallelization of the design presented, and implementing community standard work arounds for large dataset data (i.e., on-line learning, stochastic/batch gradient descent methods, k-d tree... etc.).

References

Bellet, A., Habrard, A., & Sebban, M. 2015, Synthesis Lectures on Artificial Intelligence and Machine Learning, 9, 1

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. 1994, Fast subsequence matching in time-series databases, vol. 23 (ACM)

Fulcher, B. D., Little, M. A., & Jones, N. S. 2013, Journal of the Royal Society Interface, 10, 20130048

Glanz, H., & Carvalho, L. 2013, arXiv preprint arXiv:1309.6609

Graham, M. J., Djorgovski, S., Mahabal, A. A., Donalek, C., & Drake, A. J. 2013, Monthly Notices of the Royal Astronomical Society, 431, 2371

Helfer, E., Smith, B., Haber, R., & A, P. 2015, Statistical Analysis of Functional Data, Tech. rep., Florida Institute of Technology

Hinners, T. A., Tat, K., & Thorp, R. 2018, The Astronomical Journal, 156, 7

Hu, J., Lu, J., Tan, Y. P., Yuan, J., & Zhou, J. 2017, IEEE Transactions on Circuits and Systems for Video Technology, PP, 1

Hu, J., Lu, J., Yuan, J., & Tan, Y.-P. 2014, in Asian Conference on Computer Vision (Springer), 252

Johnston, K. B., & Peter, A. M. 2017, New Astronomy, 50, 1

Mahabal, A., Sheth, K., Gieseke, F., Pai, A., Djorgovski, S. G., Drake, A., & Graham, M. 2017, in Computational Intelligence (SSCI), 2017 IEEE Symposium Series on (IEEE), 1

Richards, J. W., Starr, D. L., Miller, A. A., Bloom, J. S., Butler, N. R., Brink, H., & Crellin-Quick, A. 2012, The Astrophysical Journal Supplement Series, 203, 32

Weinberger, K. Q., Blitzer, J., & Saul, L. K. 2006, in Advances in neural information processing systems, 1473