Journal of Machine Learning Research 22 (2021) 1-7 Submitted 12/20; Revised 3/21; Published 5/21

mvlearn: Multiview Machine Learning in Python

Ronan Perry[1] RPERRY27@JHU.EDU
Gavin Mischler[9] GM2944@COLUMBIA.EDU
Richard Guo[2] RICHARDG7890@GMAIL.COM
Theodore Lee[1] TLEE124@JHU.EDU
Alexander Chang[1] ALEXC3071@GMAIL.COM
Arman Koul[1] ARMANKOUL@GMAIL.COM
Cameron Franz[2] CFRANZ3@JHU.EDU
Hugo Richard[5] HUGO.RICHARD@INRIA.FR
Iain Carmichael[6] IDC9@UW.EDU
Pierre Ablin[7] PIERRE.ABLIN@ENS.FR
Alexandre Gramfort[5] ALEXANDRE.GRAMFORT@INRIA.FR
Joshua T. Vogelstein[1,3,4,8⋆] JOVO@JHU.EDU

[1] *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218*
[2] *Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218*
[3] *Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218*
[4] *Kavli Neuroscience Discovery Institute, Institute for Computational Medicine, Johns Hopkins University, Baltimore, MD 21218*
[5] *Université Paris-Saclay, Inria, Palaiseau, France*
[6] *Department of Statistics, University of Washington, Seattle, WA 98195*
[7] *CNRS and DMA, École Normale Supérieure, PSL University, Paris, France*
[8] *Progressive Learning*
[9] *Department of Electrical Engineering, Columbia University, New York, NY 10027*
⋆ *Corresponding author*

Editor: Joaquin Vanschoren

Abstract

As data are generated more and more from multiple disparate sources, multiview data sets, where each sample has features in distinct views, have grown in recent years. However, no comprehensive package exists that enables non-specialists to use these methods easily. `mvlearn` is a Python library which implements the leading multiview machine learning methods. Its simple API closely follows that of `scikit-learn` for increased ease-of-use. The package can be installed from Python Package Index (PyPI) and the `conda` package manager and is released under the MIT open-source license. The documentation, detailed examples, and all releases are available at `https://mvlearn.github.io/`.

Keywords: multiview, machine learning, python, multi-modal, multi-table, multi-block

1. Introduction

Multiview data (sometimes referred to as multi-modal, multi-table, or multi-block data), in which each sample is represented by multiple views of distinct features, are often seen

in real-world data and related methods have grown in popularity. A view is defined as a partition of the complete set of feature variables (Xu et al., 2013). Depending on the domain, these views may arise naturally from unique sources, or they may correspond to subsets of the same underlying feature space. For example, a doctor may have an MRI scan, a CT scan, and the answers to a clinical questionnaire for a diseased patient. However, classical methods for inference and analysis are often poorly suited to account for multiple views of the same sample, since they cannot properly account for complementing views that hold differing statistical properties (Zhao et al., 2017). To deal with this, many multiview learning methods have been developed to take advantage of multiple data views and produce better results in various tasks (Sun, 2013; Hardoon et al., 2004; Chao et al., 2017; Yang et al., 2014).

Although multiview learning techniques are increasingly seen in the literature, no open-source Python package exists which implements an extensive variety of methods. The most relevant existing package, `multiview` (Kanaan-Izquierdo et al., 2019), only includes 3 algorithms with an inconsistent API. `mvlearn` fills this gap with a wide range of well-documented algorithms that address multiview learning in different areas, including clustering, semi-supervised classification, supervised classification, and joint subspace learning. Additionally, `mvlearn` preprocessing tools can be used to generate multiple views from a single original data matrix, expanding the use-cases of multiview methods and potentially improving results over typical single-view methods with the same data (Sun, 2013). Subsampled sets of features have notably led to successful ensembles of independent single-view algorithms (Ho, 1998) but can also be taken advantage of jointly by multiview algorithms to reduce variance in unsupervised dimensionality reduction (Foster et al., 2008) and improve supervised task accuracy (Nigam and Ghani, 2000). The last column of Table 1 details which methods may be useful on single-view data after feature subsampling. `mvlearn` has been tested on Linux, Mac, and PC platforms, and adheres to strong code quality principles. Continuous integration ensures compatibility with past versions, `PEP8` style guidelines keep the source code clean, and unit tests provide over 95% code coverage at the time of release.

2. API Design

The API closely follows that of `scikit-learn` (Pedregosa et al., 2011) to make the package accessible to those with even basic knowledge of machine learning in Python (Buitinck et al., 2013). The main object type in `mvlearn` is the estimator object, which is modeled after `scikit-learn`'s estimator. `mvlearn` changes the familiar method `fit(X, y)` into a multiview equivalent, `fit(Xs, y)`, where `Xs` is a list of data matrices, corresponding to a set of views with matched samples (i.e. the i^{th} row of each matrix represents the features of the same i^{th} sample across views). Note that `Xs` need not be a third-order tensor as each view need not have the same number of features. As in `scikit-learn`, classes which make a prediction implement `predict(Xs)`, or `fit_predict(Xs, y)` if the algorithm requires them to be performed jointly, where the labels `y` are only used in supervised algorithms. Similarly, all classes which transform views, such as all the embedding methods, implement `transform(Xs)` or `fit_transform(Xs, y)`.

Module	Algorithm (Reference)	Maximum Views	Useful on Constructed Data from a Single Original View
Decomposition	AJIVE (Feng et al., 2018)	2	✗
Decomposition	Group PCA/ICA (Calhoun et al., 2001)	≥ 2	✗
Decomposition	Multiview ICA (Richard et al., 2020)	≥ 2	✗
Cluster	MV K-Means (Bickel and Scheffer, 2004)	2	✓
Cluster	MV Spherical K-Means (Bickel and Scheffer, 2004)	2	✓
Cluster	MV Spectral Clustering (Kumar and Daumé, 2011)	≥ 2	✓
Cluster	Co-regularized MV Spectral Clustering (Kumar et al., 2011)	≥ 2	✓
Semi-supervised	Co-training Classifier (Blum and Mitchell, 1998)	2	✓
Semi-supervised	Co-training Regressor (Zhou and Li, 2005)	2	✓
Embed	CCA (Hotelling, 1936)	2	✗
Embed	Multi CCA (Tenenhaus and Tenenhaus, 2011)	≥ 2	✗
Embed	Kernel Multi CCA (Hardoon et al., 2004)	≥ 2	✗
Embed	Deep CCA (Andrew et al., 2013)	2	✗
Embed	Generalized CCA (Afshin-Pour et al., 2012)	≥ 2	✗
Embed	MV Multi-dimensional Scaling (MVMDS) (Trendafilov, 2010)	≥ 2	✗
Embed	Omnibus Embed (Levin et al., 2017)	≥ 2	✗
Embed	Split Autoencoder (Wang et al., 2015)	2	✗

Table 1: Multiview (MV) algorithms offered in `mvlearn` and their properties.

3. Library Overview

`mvlearn` includes a wide breadth of method categories and ensures that each offers enough depth so that users can select the algorithm that best suits their data. The package is organized into the modules listed below which includes the multiview algorithms in Table 1 as well as various utility and preprocessing functions. The modules' summaries describe their use and fundamental application.

Decomposition: `mvlearn` implements the Angle-based Joint and Individual Variation Explained (AJIVE) algorithm (Feng et al., 2018), an updated version of the JIVE algorithm (Lock et al., 2013). This was originally developed to deal with genomic data and characterize similarities and differences between data sets. `mvlearn` also implements multiview independent component analysis (ICA) methods (Calhoun et al., 2001; Richard et al., 2020), originally developed for fMRI processing.

Cluster: `mvlearn` contains multiple algorithms for multiview clustering, which can better take advantage of multiview data by using unsupervised adaptations of co-training. Even when the only apparent distinction between views is the data type

of certain features, such as categorical and continuous variables, multiview clustering has been very successful (Chao et al., 2017).

Semi-supervised: Semi-supervised classification (which includes fully-supervised classification as a special case) is implemented with the co-training framework (Blum and Mitchell, 1998), which uses information from complementary views of (potentially) partially labeled data to train a classification system. If desired, the user can specify nearly any type of classifier for each view, specifically any `scikit-learn`-compatible classifier which implements a `predict_proba` method. Additionally, the package offers semi-supervised regression (Zhou and Li, 2005) using the co-training framework.

Embed: `mvlearn` offers an extensive suite of algorithms for learning latent space embeddings and joint representations of views. One category is canonical correlation analysis (CCA) methods, which learn transformations of two views such that the outputs are highly correlated. Many software libraries include basic CCA, but `mvlearn` also implements several more general variants, including multiview CCA (Tenenhaus and Tenenhaus, 2011) for more than two views, Kernel multiview CCA (Hardoon et al., 2004; Bach and Jordan, 2003; Kuss and Graepel, 2003), Deep CCA (Andrew et al., 2013), and Generalized CCA (Afshin-Pour et al., 2012) which is efficiently parallelizable to any number of views. Several other methods for dimensionality reduction and joint subspace learning are included as well, such as multiview multidimensional scaling (Trendafilov, 2010), omnibus embedding (Levin et al., 2017), and a split autoencoder (Wang et al., 2015).

Compose: Several functions for integrating single-view and multiview methods are implemented, facilitating operations such as preprocessing, merging, or creating multiview data sets. If the user only has a single view of data, view-generation algorithms in this module such as Gaussian random projections and random subspace projections allow multiview methods to still be applied and may improve upon results from single-view methods (Sun, 2013; Nigam and Ghani, 2000; Ho, 1998).

Data sets and Plotting: A synthetic multiview data generator as well as dataloaders for the Multiple Features Data Set (Breukelen et al., 1998) in the UCI repository (Dua and Graff, 2017) and the genomics Nutrimouse data set (Martin et al., 2007) are included. Also, plotting tools extend `matplotlib` and `seaborn` to facilitate visualizing multiview data.

4. Conclusion

`mvlearn` introduces an extensive collection of multiview learning tools, enabling anyone to readily access and apply such methods to their data. As an open-source package, `mvlearn` welcomes contributors to add new desired functionality to further increase its applicability and appeal. As data are generated from more diverse sources and the use of machine learning extends to new fields, multiview learning techniques will be more useful to effectively extract information from real-world data sets. With these methods accessible to non-specialists, multiview learning algorithms will be able to improve results in academic and industry applications of machine learning.

Author Contribution

Ronan Perry, Gavin Mischler — Conceptualization, API development, writing (initial draft), prototype codebase. Richard Guo, Theodore Lee, Alexander Chang, Arman Koul, Cameron Franz — Conceptualization, API development, prototype codebase. Hugo Richard, Pierre Ablin, Alexandre Gramfort — API updates, major code revisions, writing (review and edits). Iain Carmichael — methodology, major code revisions, writing (review). Joshua T. Vogelstein — Conceptualization, supervision, funding acquisition, methodology, writing (review and edits).

Acknowledgements

This work is supported by the Defense Advanced Research Projects Agency (DARPA) Lifelong Learning Machines program through contract FA8650-18-2-7834, by R01 AG066184/AG/NIA NIH HHS/United States, and through funding from Microsoft Research. We thank the NeuroData Design class and the NeuroData lab at Johns Hopkins University for support and guidance as well as the reviewers for their helpful feedback.

References

Babak Afshin-Pour, Gholam-Ali Hossein-Zadeh, Stephen C Strother, and Hamid Soltanian-Zadeh. Enhancing reproducibility of fMRI statistical maps using generalized canonical correlation analysis in NPAIRS framework. *NeuroImage*, 60(4):1970–1981, 2012.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Conference on International Conference on Machine Learning*, volume 28, pages 1247–1255. JMLR.org, 2013.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2003.

Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *IEEE International Conference on Data Mining*, page 19–26. IEEE Computer Society, 2004.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory*, pages 92–100. Association for Computing Machinery, 1998.

Martijn. Breukelen, Robert P. W. Duin, David M. J. Tax, and J. E. den Hartog. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Vince D Calhoun, Tulay Adali, Godfrey D Pearlson, and James J Pekar. A method for making group inferences from functional mri data using independent component analysis. *Human Brain Mapping*, 14(3):140–151, 2001.

Guoqing Chao, Shiliang Sun, and J. Bi. A survey on multi-view clustering. *arXiv preprint*, arXiv:1712.06246, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Qing Feng, Meilei Jiang, Jan Hannig, and J.S. Marron. Angle-based joint and individual variation explained. *Multivariate Analysis*, 166:241 – 265, 2018.

Dean P. Foster, Sham M. Kakade, and Zhang Tong. Multi-view dimensionality reduction via canonical correlation analysis. Technical report, Toyota Technical Institute-Chicago, 2008. URL `https://repository.upenn.edu/statistics_papers/150`.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Samir Kanaan-Izquierdo, Andrey Ziyatdinov, Maria Araceli Burgueño, and Alexandre Perera-Lluna. Multiview: a software package for multiview pattern recognition methods. *Bioinformatics*, 35(16):2877–2879, 2019.

Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *International Conference on Machine Learning*, page 393–400. Omnipress, 2011.

Abhishek Kumar, Piyush Rai, and Hal Daumé. Co-regularized multi-view spectral clustering. In *International Conference on Neural Information Processing Systems*, page 1413–1421. Curran Associates Inc., 2011.

Malte Kuss and Thore Graepel. The Geometry Of Kernel Canonical Correlation Analysis. Technical Report 108, Max Planck Institute for Biological Cybernetics, May 2003.

Keith Levin, Avanti Athreya, Minh Tang, Vince Lyzinski, Youngser Park, and Carey E. Priebe. A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *IEEE International Conference on Data Mining Workshops*, pages 964–967, 2017.

Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The Annals of Applied Statistics*, 7(1):523–542, 2013.

Pascal G. P. Martin, Hervé Guillou, Frédéric Lasserre, Sébastien Déjean, Annaig Lan, Jean-Marc Pascussi, Magali SanCristobal, Philippe Legrand, Philippe Besse, and Thierry Pineau. Novel aspects of ppar α-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. *Hepatology*, 45(3):767–777, 2007.

Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Conference on Information and Knowledge Management*, pages 86–93. Association for Computing Machinery, 2000.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hugo Richard, Luigi Gresele, Aapo Hyvärinen, Bertrand Thirion, Alexandre Gramfort, and Pierre Ablin. Modeling shared responses in neuroimaging studies through multiview ica. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

Arthur Tenenhaus and Michel Tenenhaus. Regularized generalized canonical correlation analysis. *Psychometrika*, 76:257–284, 2011.

Nickolay T Trendafilov. Stepwise estimation of common principal components. *Computational Statistics & Data Analysis*, 54(12):3446–3457, 2010.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, volume 37, page 1083–1092. JMLR.org, 2015.

Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint*, arXiv:1304.5634, 2013.

Yuhao Yang, Chao Lan, Xiaoli Li, Bo Luo, and Jun Huan. Automatic social circle detection using multi-view clustering. In *ACM International Conference on Conference on Information and Knowledge Management*, pages 1019–1028, 2014.

Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43 – 54, 2017.

Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *International Joint Conference on Artificial Intelligence*, page 908–913. Morgan Kaufmann Publishers Inc., 2005.