The Journal of Open Source Software

CCA-Zoo: A collection of Regularized, Deep Learning based, Kernel, and Probabilistic CCA methods in a scikit-learn style framework

James Chapman[1] **and Hao-Ting Wang**[2, 3]

1 Centre for Medical Image Computing, University College London, London, UK **2** Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal, Université de Montréal, Montréal, QC, Canada **3** Centre de Recherche de l'Hôpital du Sacré Coeur de Montréal, Université de Montréal, Montréal, QC, Canada

DOI: 10.21105/joss.03823

Software

- Review
- Repository
- Archive

Editor: Elizabeth DuPre

Reviewers:

- @robbisg
- @hugorichard
- @ejolly

Submitted: 04 October 2021
Published: 18 December 2021

Summary

Multi-view data has gained visibility in scientific research. Examples include different languages in natural language processing, as well as neuroimaging, multiomics and audiovisual data. Canonical Correlation Analysis (CCA) (Hotelling, 1992) and Partial Least Squares (PLS) are classical methods for investigating and quantifying multivariate relationships between these views of data. The goal of CCA and its variants is to find projections (and associated weights) for each view of the data into a latent space where they are highly correlated.

The original CCA is constrained by the sample-to-feature ratio. The algorithm cannot produce a solution when the number of features in one view exceeds the number of samples. To overcome this restriction, the original CCA has been developed into a family of models which include regularised (Vinod, 1976), kernelized (Hardoon et al., 2004), probabilistic/generative (Bach & Jordan, 2005), and deep learning based (Andrew et al., 2013) variants. In particular these variations have allowed practitioners to apply these models to complex, high dimensional data. Similarly, variants of PLS have been proposed including the widely used Penalized Matrix Decomposition algorithm (Witten et al., 2009) which induces sparsity in the weight vectors for interpretability and generalisation.

`cca-zoo` is a Python package that implements many variants in a simple API with standardised outputs. We would like to highlight the unique benefits our package brings to the community in comparison to other established Python packages containing implementations of CCA. Firstly, `cca-zoo` contains a number of regularised CCA and PLS for high dimensional data that have previously only been available in installable packages in R. Native Python implementation will give Python users convenient access to these powerful models for both application and the development of new algorithms. Secondly,`cca-zoo` contains several deep CCA variants written in PyTorch (Paszke et al., 2019). We adopted a modular style allowing users to apply their desired neural network architectures for each view for their own training pipeline. Thirdly, `cca-zoo` contains generative models including probabilistic and deep variational CCA. This class of variations can be used to model the multiview data generation process and even generate new synthetic samples. Finally, `cca-zoo` provides data simulation utilities to synthesize data containing specified correlation structures as well as the paired MNIST data commonly used as a toy dataset in deep multiview learning.

The Journal of Open Source Software

Statement of need

The Python ecosystem for multiview learning currently provides a few options for implementing CCA and PLS models. `scikit-learn` (Pedregosa et al., 2011) contains standard implementations of both CCA and PLS for two-view data which plug into their mature API. `pyrcca` (Bilenko & Gallant, 2016) contains implementations of ridge regularised and kernelized two-view CCA. The embed module of `mvlearn` (Perry et al., 2020) is perhaps the closest relative of `cca-zoo`, containing implementations of ridge regularised and kernelized multi-view CCA. `cca-zoo` builds on the `mvlearn` API by providing an additional range of regularised models and in particular sparsity inducing models which have found success in multiomics. Building on the reference implementation in `mvlearn`, `cca-zoo` further provides a number of deep learning models with a modular design to enable users to supply their own choice of neural network architectures.

Standard implementations of state-of-the-art models help as benchmarks for methods development and easy application to new datasets. `cca-zoo` extends the existing ecosystem with a number of sparse regularised CCA models. These variations have found popularity in genetics and neuroimaging where signals are contained in a small subset of variables. With applications like these in mind, `cca-zoo` simplified the access to the learnt model weights to perform further analysis in the feature space. Furthermore, the modular implementations of deep CCA and its multiview variants allow the user to focus on architecture tuning. Finally, `cca-zoo` adds generative models including variational (C. Wang, 2007) and deep variational CCA (W. Wang et al., 2016) as well as higher order canonical correlation analysis with tensor (Kim et al., 2007) and deep tensor CCA (Wong et al., 2021).

Implementation

`cca-zoo` adopted a similar API to that used in `scikit-learn`. The user first instantiates a model object and its relevant hyperparameters. Next they call the model's `fit()` method to apply the data. After fitting, the model object contains its relevant parameters such as weights or dual coefficients (for kernel methods) which can be accessed for further analysis. For models that fit with iterative algorithms, the model may also contain information about the convergence of the objective function. After the model has been fit, its `transform()` method can project views into latent variables and `score()` can be used to measure the canonical correlations.

The deep and probabilistic models are supported by PyTorch and NumPyro respectively. Due to the size of these dependencies, these two classes of variations are not in the default installation. Instead, we provide options [deep] and [probabilistic] for users. The list bellow provides the complete collection of models along with their installation tag is provided below.

Model List

A complete model list at the time of publication:

Model Class	Model Name	Number of Views	Install
CCA	Canonical Correlation Analysis	2	standard
rCCA	Canonical Ridge	2	standard
KCCA	Kernel Canonical Correlation Analysis	2	standard
MCCA	Multiset Canonical Correlation Analysis	>=2	standard

Model Class	Model Name	Number of Views	Install
KMCCA	Kernel Multiset Canonical Correlation Analysis	>=2	standard
GCCA	Generalized Canonical Correlation Analysis	>=2	standard
KGCCA	Kernel Generalized Canonical Correlation Analysis	>=2	standard
PLS	Partial Least Squares	>=2	standard
CCA_ALS	Canonical Correlation Analysis by Alternating Least Squares) (Golub & Zha, 1995)	>=2	standard
PLS_ALS	Partial Least Squares by Alternating Least Squares)	>=2	standard
PMD	Sparse CCA by Penalized Matrix Decomposition	>=2	standard
ElasticCCA	Sparse Penalized CCA (Waaijenborg et al., 2008)	>=2	standard
ParkhomenkoCCA	Sparse CCA (Parkhomenko et al., 2009)	>=2	standard
SCCA	Sparse Canonical Correlation Analysis by Iterative Least Squares (Mai & Zhang, 2019)	>=2	standard
SCCA_ADMM	Sparse Canonical Correlation Analysis by Altnerating Direction Method of Multipliers (Suo et al., 2017)	>=2	standard
SpanCCA	Sparse Diagonal Canonical Correlation Analysis (Asteris et al., 2016)	>=2	standard
SWCCA	Sparse Weighted Canonical Correlation Analysis (Wenwen et al., 2018)	>=2	standard
TCCA	Tensor Canonical Correlation Analysis	>=2	standard
KTCCA	Kernel Tensor Canonical Correlation Analysis (Kim et al., 2007)	>=2	standard
DCCA	Deep Canonical Correlation Analysis	>=2	deep
DCCA_NOI	Deep Canonical Correlation Analysis by Non-Linear Orthogonal Iterations (W. Wang, Arora, Livescu, & Srebro, 2015)	>=2	deep

Model Class	Model Name	Number of Views	Install
DCCAE	Deep Canonically Correlated Autoencoders (W. Wang, Arora, Livescu, & Bilmes, 2015)	>=2	deep
DTCCA	Deep Tensor Canonical Correlation Analysis	>=2	deep
SplitAE	Split Autoencoders (Ngiam et al., 2011)	2	deep
DVCCA	Deep Variational Canonical Correlation Analysis	>=2	deep
ProbabilisticCCA	Probabilistic Canonical Correlation Analysis	2	probabilistic

Documentation

The package is accompanied by documentation (https://cca-zoo.readthedocs.io/en/latest/index.html) and a number of tutorial notebooks which serve as both guides to the package as well as educational resources for CCA and PLS methods.

Conclusion

`cca-zoo` fills many of the gaps in the multiview learning ecosystem in Python, including a flexible API for deep-learning based models, regularised models for high dimensional data (and in particular those that induce sparsity), and generative models.`cca-zoo` will therefore help researchers to apply and develop Canonical Correlation Analysis and Partial Least Squares models. We continue to welcome contributions from the community.

Acknowledgements

JC is supported by the EPSRC-funded UCL Centre for Doctoral Training in Intelligent, Integrated Imaging in Healthcare (i4health) (EP/S021930/1) and the Department of Health's NIHR-funded Biomedical Research Centre at University College London Hospitals. HTW is supported by funds from la Fondation Courtois awarded to Dr. Pierre Bellec.

References

Andrew, G., Arora, R., Bilmes, J., & Livescu, K. (2013). Deep canonical correlation analysis. *International Conference on Machine Learning*, 1247–1255.

Asteris, M., Kyrillidis, A., Koyejo, O., & Poldrack, R. (2016). A simple and provable algorithm for sparse diagonal CCA. *International Conference on Machine Learning*, 1148–1157.

Asteris, M., Kyrillidis, A., Koyejo, O., & Poldrack, R. (2016). A simple and provable algorithm for sparse diagonal CCA. *International Conference on Machine Learning*, 1148–1157.

Bach, F. R., & Jordan, M. I. (2005). *A probabilistic interpretation of canonical correlation analysis*. https://statistics.berkeley.edu/sites/default/files/tech-reports/688.pdf

Bilenko, N. Y., & Gallant, J. L. (2016). Pyrcca: Regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in Neuroinformatics*, *10*, 49. https://doi.org/10.3389/fninf.2016.00049

Golub, G. H., & Zha, H. (1995). The canonical correlations of matrix pairs and their numerical computation. In *Linear algebra for signal processing* (pp. 27–49). Springer. https://doi.org/10.1007/978-1-4612-4228-4_3

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, *16*(12), 2639–2664. https://doi.org/10.1162/0899766042321814

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics* (pp. 162–190). Springer. https://doi.org/10.2307/2333955

Kim, T.-K., Wong, S.-F., & Cipolla, R. (2007). Tensor canonical correlation analysis for action classification. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. https://doi.org/10.1109/cvpr.2007.383137

Mai, Q., & Zhang, X. (2019). An iterative penalized least squares approach to sparse canonical correlation analysis. *Biometrics*, *75*(3), 734–744. https://doi.org/10.1111/biom.13043

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. *ICML*.

Parkhomenko, E., Tritchler, D., & Beyene, J. (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, *8*(1). https://doi.org/10.2202/1544-6115.1406

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8026–8037.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Perry, R., Mischler, G., Guo, R., Lee, T., Chang, A., Koul, A., Franz, C., Richard, H., Carmichael, I., Ablin, P., & others. (2020). Mvlearn: Multiview machine learning in python. *arXiv Preprint arXiv:2005.11890*.

Suo, X., Minden, V., Nelson, B., Tibshirani, R., & Saunders, M. (2017). Sparse canonical correlation analysis. *arXiv Preprint arXiv:1705.10865*.

Vinod, H. D. (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, *4*(2), 147–166. https://doi.org/10.1016/0304-4076(76)90010-5

Waaijenborg, S., Witt Hamer, P. C. V. de, & Zwinderman, A. H. (2008). Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology*, *7*(1). https://doi.org/10.2202/1544-6115.1329

Wang, C. (2007). Variational bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, *18*(3), 905–910. https://doi.org/10.1109/tnn.2007.891186

Wang, W., Arora, R., Livescu, K., & Bilmes, J. (2015). On deep multi-view representation learning. *International Conference on Machine Learning*, 1083–1092.

Wang, W., Arora, R., Livescu, K., & Srebro, N. (2015). Stochastic optimization for deep CCA via nonlinear orthogonal iterations. *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (allerton)*, 688–695. https://doi.org/10.1109/allerton.2015.7447071

Wang, W., Yan, X., Lee, H., & Livescu, K. (2016). Deep variational canonical correlation analysis. *arXiv Preprint arXiv:1610.03454*.

Wenwen, M., Juan, L., & Zhang, S. (2018). Sparse weighted canonical correlation analysis. *Chinese Journal of Electronics*, *27*(3), 459–466. https://doi.org/10.1049/cje.2017.08.004

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534. https://doi.org/10.1093/biostatistics/kxp008

Wong, H. S., Wang, L., Chan, R., & Zeng, T. (2021). Deep tensor CCA for multi-view learning. *IEEE Transactions on Big Data*.