Independent Vector Analysis for Data Fusion Prior to Molecular Property Prediction with Machine Learning

Zois Boukouvalas
University of Maryland, College Park
Dept. of Mechanical Engineering
College Park, MD 20742
zoisb@umd.edu

Daniel C. Elton
University of Maryland, College Park
Dept. of Mechanical Engineering
College Park, MD 20742
delton@umd.edu

Peter W. Chung
University of Maryland, College Park
Dept. of Mechanical Engineering
College Park, MD 20742
pchung15@umd.edu

Mark D. Fuge
University of Maryland, College Park
Dept. of Mechanical Engineering
College Park, MD 20742
fuge@umd.edu

November 5, 2018

Abstract

Due to its high computational speed and accuracy compared to ab-initio quantum chemistry and forcefield modeling, the prediction of molecular properties using machine learning has received great attention in the fields of materials design and drug discovery. A main ingredient required for machine learning is a training dataset consisting of molecular features—for example fingerprint bits, chemical descriptors, etc. that adequately characterize the corresponding molecules. However, choosing features for any application is highly non-trivial. No "universal" method for feature selection exists. In this work, we propose a data fusion framework that uses Independent Vector Analysis to exploit underlying complementary information contained in different molecular featurization methods, bringing us a step closer to automated feature generation. Our approach takes an arbitrary number of individual feature vectors and automatically generates a single, compact (low dimensional) set of molecular features that can be used to enhance the prediction performance of regression models. At the same time our methodology retains the possibility of interpreting the generated features to discover relationships between molecular structures and properties. We demonstrate this on the QM7b dataset for the prediction of several properties such as atomization energy, polarizability, frontier orbital eigenvalues, ionization potential, electron affinity, and excitation energies. In addition, we show how our method helps improve the prediction of experimental binding affinities for a set of human BACE-1 inhibitors. To encourage more widespread use of IVA we have developed the PyIVA Python package, an open source code which is available for download on Github.

1 Introduction

Machine learning (ML) has recently been used for the prediction of molecular properties and studies have shown that it can provide accurate and computationally efficient solutions for this task [1, 2, 3, 4, 5, 6, 7]. The prediction ability of a ML model highly depends on the proper selection of a training dataset, via feature vectors, that can fully capture certain characteristics of a given set of molecules. A common way to represent a molecule is through a string representation called the Simplified Molecular-Input Line-Entry System (SMILES) string [8]. Although working directly with SMILES strings has shown to be effective in some ML tasks [9], most of the ML methods require vector or matrix variables, rather than strings. Basic classes of featurization methods that have been widely used in the literature include cheminformatic descriptors, molecular fingerprints, and custom graph convolution based fingerprints, where each featurization method provides different – though not necessarily unique – information about a molecule.

Thus it is important to find answers to the question "how can disparate datasets, each associated with unique featurization methods, be integrated?" The question is motivated by the desire to create automated approaches where data generation techniques are integrable with ML models. *Data fusion* methods [10, 11] may serve this purpose since they enable simultaneous study of multiple datasets by, for instance, exploiting alignments of data fragments where there is a common underlying latent space. A naive approach could be to simply concatenate data that has been generated by different featurization methods. However, this typically leads to a curse of dimensionality that could affect the performance of a ML algorithm and make it impossible to discover the features of greatest importance. Therefore, selecting a model that generates a set of molecular feature vectors by effectively exploiting complementary information among multiple datasets is an important issue.

Blind source separation (BSS) techniques enable the joint analysis of datasets and extraction of summary factors with few assumptions placed on the data. This is generally achieved through the use of a generative model. One of the most widely used BSS techniques is independent component analysis (ICA) [12, 13]. The popularity is likely because by only requiring statistical independence of the latent sources it can uniquely identify the true latent sources subject to only scaling and permutation ambiguities. However, ICA can only decompose a single dataset. In many applications multiple sets of data are gathered about the same phenomenon. These sets can often be disjoint and merely misaligned fragments of a shared underlying latent space. Multiple datasets could therefore share dependencies. This motivates the application of methods that can jointly factorize multiple datasets, like independent vector analysis (IVA) [14]. IVA is a recent generalization of ICA to multiple datasets that can achieve improved performance over performing ICA on each dataset separately.

In this paper, we propose a novel data fusion framework that uses IVA to exploit underlying complementary information contained across multiple datasets. Namely, we show that information that has been generated by different molecular featurization methods can be fused to improve learning of the chemical relationships. Note here that achieving perfect regression error is not the the goal of this work. Instead, our goal is to determine how to generate feature vectors by combining datasets from multiple featurization methods to improve the learned response of the data-fused model over the performance of the individual feature vectors treated separately. Particularly noteworthy is that the proposed approach is parameter free, computationally attractive when compared with existing methods [5], easily interpretable due to the simplicity of the generative model, and does not require a large amount of training samples in order to achieve a desirable regression error.

The remainder of this paper is organized as follows. In Section 2, we provide a brief background on the IVA method and the regression procedure. Section 3 provides the results of the regression procedure and associated discussions. The conclusions and future research directions are presented in Section 4.

2 Materials and Methods

The BSS model is formulated as follows. Let $\mathbf{X} \in \mathbb{R}^{d \times N}$ be the observation matrix where d denotes the dimension of the feature vector and N denotes the total number of molecules. The noiseless BSS generative model is given by

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{d \times P}$ is the mixing matrix, and $\mathbf{S} \in \mathbb{R}^{P \times N}$ is the matrix that contains the sources that need to be estimated and will be used as the new feature vector for the ML task. One of the most widely used methods for solving the BSS problem (1) is ICA and its basic assumption is that the source signals are statistically independent [15, 16, 17, 18, 19].

By rewriting (1) using random vector notation, we have

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n), \quad n = 1, \ldots, N, \tag{2}$$

where n is the sample index denoting the nth molecule, $\mathbf{s}(n) \in \mathbb{R}^P$ are the unknown sources that need to be estimated, and $\mathbf{x}(n) \in \mathbb{R}^d$ are the mixtures. Our interest is in dealing with overdetermined problems where $d > P$. This can be reduced to the case where $d = P$ using a dimensionality reduction technique like principal component analysis (PCA). For the purpose of this work and for the rest of this paper we assume that the samples of each unknown source are independent and identically distributed and therefore, to simplify the notation we drop the sample index n.

Although ICA has been shown to be very useful in many applications, it decomposes a single dataset at a time. For practical applications that involve more than one dataset one could perform ICA separately on each dataset and align the subsequent results. However, this approach could be considered suboptimal since performing ICA individually on each dataset will ignore any dependencies that exist among them.

2.1 Independent Vector Analysis

It is common for datasets that have been generated from different featurization methods to have some inherent dependence among them. IVA generalizes the ICA problem by allowing for full exploitation of this dependence leading to improved performance beyond what is achievable by applying ICA separately to each dataset. Additionally, IVA automatically aligns dependent sources across the datasets, thus bypassing the need for a second permutation correction algorithm.

IVA is similar to ICA, except that now we have K datasets, $\mathbf{x}^{[k]}$, $k = 1, ..., K$ where each dataset is a linear mixture of N statistically independent sources. Under the assumption that PCA preprocessing results in $d = P$, the noiseless IVA model is given by

$$\mathbf{x}^{[k]} = \mathbf{A}^{[k]}\mathbf{s}^{[k]}, \quad k = 1, ..., K, \tag{3}$$

where $\mathbf{A}^{[k]} \in \mathbb{R}^{P\times P},\ k = 1, ..., K$ are invertible mixing matrices and $\mathbf{s}^{[k]} = [s_1^{[k]}, ..., s_P^{[k]}]^\top$ is the vector of latent sources for the kth dataset. In the IVA model, the components within each $\mathbf{s}^{[k]}$ are assumed to be independent, while at the same time, dependence across corresponding components of $\mathbf{s}^{[k]}$ is allowed. To mathematically formulate the dependence across components that IVA can take into account, we define the source component vector (SCV) by vertically concatenating the pth source from each of the K dataset as

$$\mathbf{s}_p = [s_p^{[1]}, ..., s_p^{[K]}]^\top, \tag{4}$$

where $\mathbf{s}_p$ is a K-dimensional random vector. The goal in IVA is to estimate K demixing matrices to yield source estimates $\mathbf{y}^{[k]} = \mathbf{W}^{[k]}\mathbf{x}^{[k]}$, such that each SCV is maximally independent of all other SCVs. It should be mentioned that dependence across datasets is not a necessary condition for IVA to work. In the case where this type of statistical property does not exist, IVA reduces to individual ICAs on each dataset.

The IVA cost function can be defined in a similar manner as ICA. However, the optimization parameter is not just a single demixing matrix $\mathbf{W}$ as in the ICA case, but a set of demixing matrices $\mathbf{W}^{[1]}, \ldots, \mathbf{W}^{[K]}$, which can be collected into a three dimensional array $\mathcal{W} \in \mathbb{R}^{P\times P\times K}$. The IVA objective function is given by

$$J_{IVA}(\mathcal{W}) = \sum_{p=1}^{P} H(\mathbf{y}_p) - \sum_{k=1}^{K} \log\left|\det\left(\mathbf{W}^{[k]}\right)\right| - H(\mathbf{x}^{[1]}, ..., \mathbf{x}^{[K]}), \tag{5}$$

where $H(\mathbf{y}_p)$ denotes the differential entropy of the estimated pth SCV and the term $H(\mathbf{x}^{[1]}, ..., \mathbf{x}^{[K]})$ is a constant parameter where it can be treated as a constant for optimization purposes. By definition, the term $H(\mathbf{y}_p)$ equals $\sum_{k=1}^{K} H(y_p^k) - I(\mathbf{y}_p)$, where $I(\mathbf{y}_p)$ denotes the mutual information within the pth SCV. Therefore, it can be observed that minimization with respect to each demixing matrix $\mathbf{W}^{[k]}$ of (5) automatically increases the mutual information within the components of a SCV, revealing how IVA exploits this type of statistical property. It can be seen that without the mutual information term, the objective function (5) is equivalent to performing independent ICA separately on each dataset. For more information about the derivation of the IVA cost function and the optimization scheme, we refer the reader to [20].

From (5) it can be observed that the probability density function or its approximation for each estimated SCV plays an important role on the estimation performance of the demixing matrices. One of the first IVA algorithms was formulated for solving the convolutive ICA problem in the frequency domain using multiple frequency bins [19]. This led to the development of IVA-Laplacian (IVA-L) [14], an algorithm that takes higher-order statistics (HOS) into account and assumes a Laplacian distribution for the underlying source component vectors. Due to its simplicity and computational efficiency, for the purpose of our work we use IVA-L.

2.2 Feature extraction and regression procedure

The regression process consists of three stages. In the first stage, N molecules are randomly sub sampled in order to generate training and testing datasets, $\mathbf{X}^k_{\text{Train}}$ and $\mathbf{X}^k_{\text{Test}}$ respectively, for each $k = 1, \ldots, K$.

In the second stage, the mean from each dataset is removed and PCA is applied to each $\mathbf{X}^k_{\text{Train}}$ using an order P, implying that the signal subspace contains the components that have higher variance. Then for each $k \in \{1, \ldots, K\}$, we generate $\hat{\mathbf{X}}^k_{\text{Train}} \in \mathbb{R}^{P\times N_{\text{Train}}}$ and by vertically concatenating each $\hat{\mathbf{X}}^k_{\text{Train}}$ we form a three dimensional array $\hat{\mathbf{X}}_{\text{Train}} \in \mathbb{R}^{P\times N\times K}$. IVA-L is performed on $\hat{\mathbf{X}}_{\text{Train}}$ resulting in a set of demixing matrices, $\{\mathbf{W}^1, \ldots, \mathbf{W}^K\}$, where each $\mathbf{W}^k \in \mathbb{R}^{P \times P}$. Using the demixing matrices we generate $\mathbf{Y}^k_{\text{Train}} = \mathbf{W}^k(\hat{\mathbf{X}}^k_{\text{Train}})^\top$ for each $k = 1 \ldots, K$ and the training dataset $\mathbf{Y}_{\text{Train}}$ is formed by vertically concatenating the SCVs, $\mathbf{y}_1, \ldots, \mathbf{y}_P$.

The testing dataset is generated by removing the mean from each testing dataset and by using the PCA training transformation from the training phase, we generate $\hat{\mathbf{X}}^k_{\text{Train}},\ k = 1 \ldots, K$. Each of the demixing matrices from the training phase is used to create testing datasets $\mathbf{Y}^k_{\text{Test}} \in \mathbb{R}^{P \times N_{\text{Test}}}$ by $\mathbf{Y}^k_{\text{Test}} = \mathbf{W}^k(\hat{\mathbf{X}}^k_{\text{Test}})^\top$ for each $k = 1 \ldots, K$. Finally, the testing dataset is formed by vertically concatenating the testing SCVs.

In the third stage, we train the regression model using $(\mathbf{Y}_{\text{Train}})^\top$. Here, the specific form of the regression function is unimportant. But to demonstrate a concrete example, we use kernel ridge regression (KRR) with a Gaussian kernel, which follows from previous work [3] and [1]. Once the regression model has been trained we evaluate its performance by using the unseen data $(\mathbf{Y}_{\text{Test}})^\top$. For all of the experiments, hyperparameter optimization and model training and testing is done using a nested cross validation scheme. In the inner loop, the length scale parameter of the Gaussian kernel and the regularization parameter are optimized using grid search selection using 80% of the data to train the model and another 10% for validation. The remaining 10% is held out as a test set to estimate performance after hyperparameter optimization. The outer loop is done five times, corresponding to five folds. This entire process was repeated 30 times (with shuffling before each iteration) to generate well-converged statistics.

2.3 Data Sources and Methods

The scope of the present work is limited to the data from two sources: the QM7b dataset [21, 5] and the human β-secretase-1 (BACE-1) inhibitors dataset from the *moleculenet.ai* website [22].

The QM7b dataset contains 7211 organic molecules made of up to 7 heavy atoms. These atoms include C, N, O, S, and Cl.The only other element present in the molecules is hydrogen. For each molecule several properties have been calculated at different levels of theory, including density functional theory and the many-body GW approach. The properties include the atomization energy, polarizability, HOMO and LUMO eigenvalues, and excitation energy. From the QM7b data, we generate three datasets using three different featurization methods: Coulomb matrices eigenspectra (CME) [1], sum over bonds (SOB) [3], and weight matrices eigenspectra (WE). The Coulomb matrix of a molecule is specified by the 3-dimensional coordinates of the atoms as well as their atomic charges. Since the Coulomb matrix is not invariant under row or column permutations, the eigenspectra are used as a feature vector. The dimensionality of the Coulomb matrix ($d = 23$ for QM7b) is set by the molecule with the largest number of atoms in the dataset. Sum over bonds is defined as a bond count and the feature vectors are generated by first enumerating all of the bond types in the dataset and then counting how many of each bond are present in each molecule ($d = 28$ for QM7b). To compute the weight matrix eigenspectra (WE) featurization, each molecule is treated as a graph, where edges represent bonds and vertices represent atoms. The value of each matrix element represents the bond order, e.g. ‘1’ denotes a single bond, ‘2’ a double bond etc. Similarly to the Coulomb matrix featurization method, for each molecule we define the feature vector to be the eigenspectrum of its weight matrix, and the dimension of the WE feature vector is $d = 23$.

The BACE-1 inhibitor dataset is obtained from the *moleculenet.ai* website.[22] 1,513 of the values were taken from an earlier work [23], which in turn obtained them from over thirty different laboratories. We then created seven basic datasets using the descriptors contained in the MoleculeNet dataset and descriptors generated with the *mmltoolkit* [24] and *RDKit* [25]. The first dataset ($d = 13$) consists of physio-chemical descriptors computed using BIOVIA’s Pipeline Pilot package. The second dataset ($d = 32$) consists of additional Canvas descriptors.[26] The third dataset ($d = 28$) uses the SOB featurization method. The fourth dataset ($d = 88$) consists of counts of each atom type as defined in the electrotopological state index (the “E-state” count fingerprint).[27]. Dataset five ($d = 34$) consists of the RDKit molecular surface charge descriptors [25, 28]. Dataset six ($d = 60$) consists of the RDKit functional group counts and dataset seven ($d = 28$) consists of an additional custom set of functional group counts obtained using the *functional_group_featurizer* in the *mmltoolkit*. We then use these basic seven sets to create $K = 7$ different combinations of sets. Therefore, the case where $K = 2$ consists of 21 different combinations from the seven basic sets, and when $K = 3$ there are 35 combinations, etc.

3 Results and Discussion

The QM7b dataset study examines a) regression performance with respect to three fusing methods and the size of the datasets and b) interpretability properties of IVA. The relatively large size of the dataset and the number of chemical properties make QM7b desirable for this part of the study. The BACE-1 inhibitor dataset is used to demonstrate how different featurization methods complement each other and to contrast the effects of different combinations of featurizations on the performance of the regression model.

Table 1: Average MAE for different featurization methods and different approaches to generate feature vectors for 14 properties.

Property	Units	SOB	CME	WE	Regular	ICA	IVA
E(PBEO)	Kcal/mol	6.029	9.966	44.398	**2.662**	3.187	2.989
$E^*_{\max}$	eV	1.206	2.076	1.65	1.423	1.478	**1.335**
$I_{\max}$	Arbitrary	0.082	0.105	0.106	0.081	0.08	**0.074**
HOMO(ZINDO)	eV	0.255	0.488	0.537	**0.183**	0.206	0.194
LUMO(ZINDO)	eV	0.181	0.57	0.529	0.124	0.111	**0.105**
E^*_{1st}(ZINDO)	ev	0.266	0.693	0.714	0.176	0.183	**0.174**
IP(ZINDO)	eV	0.279	0.507	0.551	**0.225**	0.246	0.229
EA(ZINDO)	eV	0.181	0.632	0.575	0.135	0.12	**0.113**
HOMO(PBEO)	eV	0.232	0.344	0.41	**0.18**	0.207	0.19
LUMO(PBEO)	eV	0.183	0.304	0.324	0.132	0.141	**0.128**
HOMO(GW)	eV	0.254	0.375	0.446	**0.195**	0.222	0.205
LUMO(GW)	eV	0.182	0.235	0.242	0.154	0.156	**0.141**
α(PBEO)	Å^3	0.224	0.308	0.472	0.126	0.136	**0.123**
α(SCS)	Å^3	0.259	0.202	0.434	**0.086**	0.112	0.1

3.0.1 Regression analysis

The first three columns in Table (1) show the MAEs for out-of-sample predictions by the KRR model for each type of featurization method. Errors are reported for the 14 properties associated with the QM7b dataset. From (1), we see that SOB provides the best results except for α(SCS), where CME is more accurate. Overall, WE performs the worst. These results are not surprising since WE contains information only about the bonds within a molecule and is the only featurization method that lacks information about the atomic species in each molecule, which is critical to many of the properties.

The last three columns of Table (1) list the MAEs when different techniques have been used to generate feature vectors for training the regression model. The so-called Regular approach denotes the procedure where the SOB, CME, and WE datasets are vertically concatenated resulting in N feature vectors of dimension 74. The ICA and IVA approaches share the same feature extraction and regression procedures described in (2.2). The only difference is that after we apply PCA to each $\mathbf{X}^k_{\text{Train}}$, we perform ICA by entropy bound minimization (ICA-EBM) [29] on each $\hat{\mathbf{X}}^k_{\text{Train}}$ training dataset separately. For both ICA and IVA approaches the resulting N feature vectors are of dimension 30. From Table (1), we observe that IVA leads to an improved MAE for 8 out of 14 properties when compared to the "Regular" approach. For all properties we have examined, IVA yields a lower MAE than ICA. This is likely due to IVA's ability to resolve separate demixing matrices for each of the K data sets, thereby exploiting the complementary information that is shared across them.

We also explored the convergence behavior of the regression model as the number of molecules in the training dataset is increased. Plots of MAE as a function of training points N are called learning curves and typically follow a power law MAE$= C * N^{\alpha}$, where C is a constant and α is the power law exponent which we refer to as the rate of convergence. These results suggest that IVA supports small data problems well. However, when large data are available, a fusion approach becomes less beneficial due to the smaller convergence rate.

The calculated rates are shown in Figure (1) for each approach and for several representative properties. We observe that IVA has a smaller rate of convergence as a function of sample size but lower absolute error through most of the range of sample sizes we examined. This indicates that the benefits of fusing data are best when the data are small and that a likely cross-over point occurs where the benefits of fusing data are outweighed by the slower convergence. On the left in Figure (1) we mark in bold the properties where IVA provides a lower MAE when the full dataset is used. On the right in Figure (1), we show the MAE obtained after five-fold cross-validation as a function of different N for four representative target properties. For the most part, absolute errors from IVA are substantially lower. The degree to which the absolute error is smaller depends on the target property and data size. For instance, the Regular approach gives the lowest MAE for atomization energy (E(PBE0)). For E^*_{1st}(ZINDO), IVA gives the lowest MAE, but we see that as N increases the gains from using IVA decrease as more data is added indicating that the the curse of dimensionality becomes less of an issue for larger N. For IP(ZINDO) the slope of both IVA and regular are approximately the same. Finally, for LUMO(GW) we see a large improvement from IVA, and ICA and the Regular approach tend to achieve the same MAE as N increases.

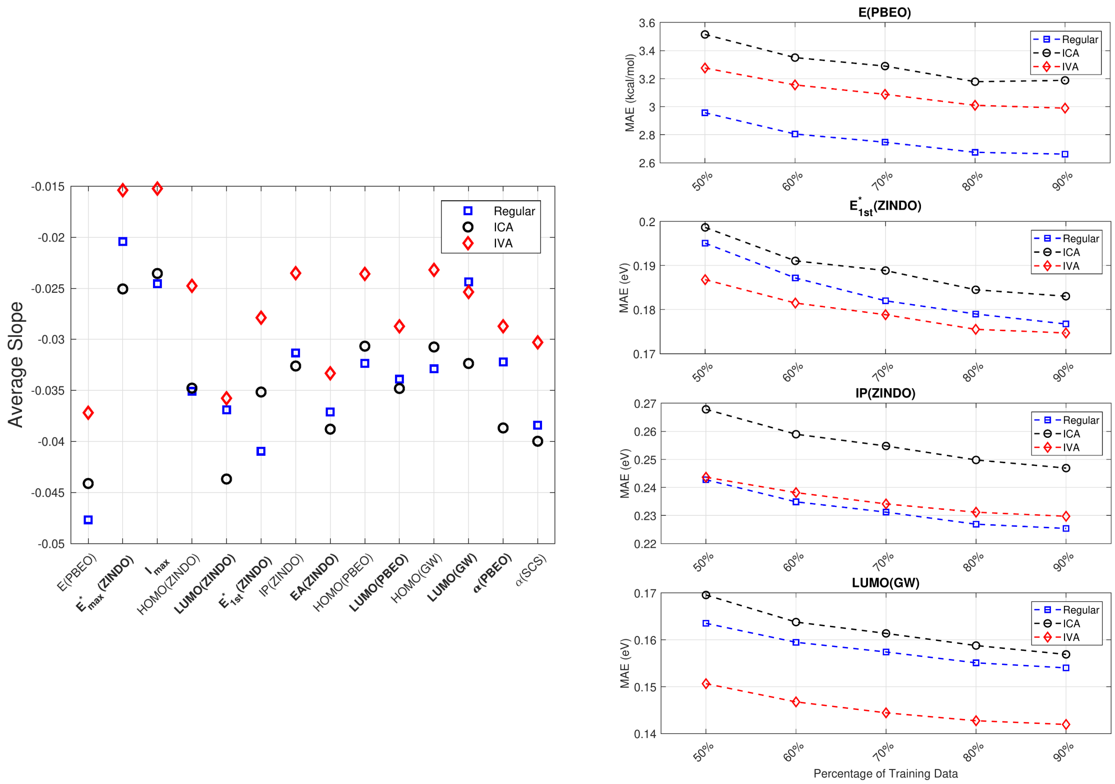

Figure 1: On the left: Slope of the learning curve on a log-log plot. On the right: Representative examples showing how the relative performance of IVA, ICA, and regular approach can be significantly different depending on the target property being predicted.

3.0.2 Significance of mixing matrices

An important property of IVA is that it can also provide chemical or physical interpretations through the estimated mixing matrix. Once the demixing matrices, $\mathbf{W}^{[k]}$ have been estimated using the approach in Section 2.2, we can estimate the mixing matrix by performing back-reconstruction. In order to describe this procedure remember that before IVA is performed, PCA is applied to each $\mathbf{X}^k$ using a presumed order P. This provides the data reduction matrix $\mathbf{F}^k$ where each $\mathbf{F}^k$ is formed by the eigenvectors with the first P highest eigenvalues of the corresponding $\mathbf{X}^k$. An estimate of the mixing matrix $\mathbf{A}^k$ is computed as $\hat{\mathbf{A}}^k = (\mathbf{F}^k)^\dagger(\mathbf{W}^k)^{-1}$, where $(\cdot)^\dagger$ denotes the pseudo-inverse of a matrix.

Each row of the kth estimated demixing matrix therefore represents the weights for the estimated sources of the kth dataset. The values of the weights can reveal relationships between certain characteristics of a given set of molecules. The weights in the mixing matrix from the SOB features are shown in Figure (2). A common motif appears for the weights associated with the CN and CC triple bond. Other motifs appear for other molecular features, which we are in the process of analyzing further. This provides early evidence that the IVA generative model encodes information regarding the relationships among different molecular features.

3.1 (BACE-1) inhibitors dataset

We next evaluate the performance of the IVA approach as a function of the number of datasets that are fused. The specific targets of the calculation are the experimental binding affinities (IC50 values) for a set of 1,522 human β-secretase-1 (BACE-1) inhibitors.

Figure (3) displays the MAE of the IC50 prediction as a function of the number of featurization methods used to train the regression model. We chose the dimension of the vectors created by IVA to be $P = 10$ for each dataset. We also excluded ICA hereafter upon observing no significant performance benefit beyond the Regular and IVA in the last section using the QM7b dataset.

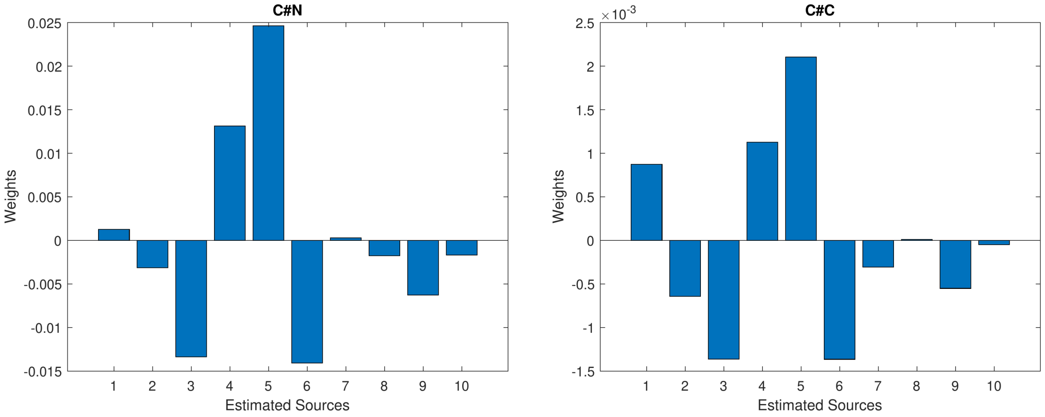

Figure 2: Weights for different features in SOB as a function of the estimated sources. Bars indicate the value of each element in the corresponding row of the estimated mixing matrix.

The MAEs in Figure (3) show the improvement in the regression error with an increasing number of fused datasets. For the sake of comparison, the errors associated with the Regular approach are also shown. When $K = 2$ the median of the Regular approach is lower than the IVA approach. However, IVA reduces the variation in the MAEs which reveals how it produces compact features. As expected, as the number of datasets is increased, IVA performs better than the Regular approach, due to the fact that IVA exploits complementary information among the different featurization methods.

In addition, Figure (3) shows that as we increase the number of featurization methods, the rate the median MAE decreases is larger than the rate using the Regular approach. Finally, note that for each case the dimension of the generated feature vectors when using the IVA approach is always lower compared to the dimension of the feature vectors in the Regular approach.

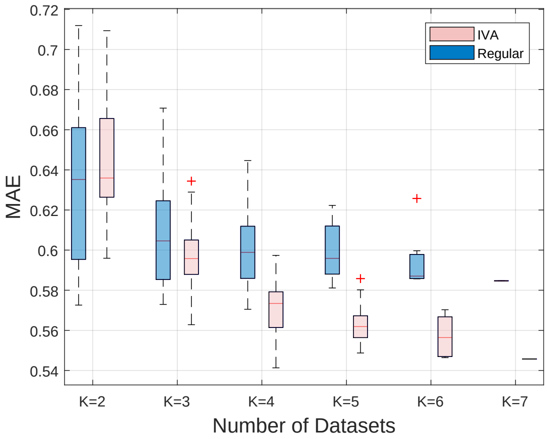

Figure 3: Boxplots of the MAE as a function of different number of datasets used in order to train the regression model. Boxplots provide statistics for the MAE when combining different featurization methods.

4 Conclusion

Different featurization methods provide different information about a molecule. Thus generating features by adopting a method that exploits the inherent dependence among those different featurization methods can benefit ML tasks. In this work, we have proposed a data fusion framework that uses joint BSS techniques to exploit the underlying complementary information contained in different featurization methods. The new framework is based on IVA, a method that effectively exploits the dependence among different featurization methods and provides feature vectors that effectively can train a regression model. The IVA method that has been used is parameter free and computationally efficient with lower complexity than existing state of the art (e.g., neural network methods). In addition, its simple generative model may enable the illumination of fundamental relationships among certain characteristics of molecules, which might be of independent scientific interest. To encourage further research and application of IVA we have written the first open source Python package for IVA, pyIVA, which is available on Github.[30] The code is based off an existing Matlab code.[31] Currently only IVA with a multivariate Laplacian prior is implemented, but we hope our work will lead to more IVA methods being implemented in Python.

The success of the proposed method raises several interesting questions that can be explored in future work. Depending on the nature of the data we can use or develop new algorithms that take different statistical properties into account such as sparsity [32]. In addition, we can also compare the regression performance with other IVA algorithms or other methods such as dictionary learning [33, 34] and non-negative matrix or tensor factorization [35]. Lastly, order selection techniques can be used to determine the order of the reduced IVA space.

Acknowledgments

Support for this work is gratefully acknowledged from the U.S. Office of Naval Research under grant number N00014-17-1-2108 and from the Energetics Technology Center under project number 2044-001. Partial support is also acknowledged from the Center for Engineering Concepts Development in the Department of Mechanical Engineering at the University of Maryland, College Park. We thank Dr. Ruth M. Doherty and Dr. Bill Wilson from the Energetics Technology Center for their encouragement, useful thoughts, and for proofreading the manuscript.

References

[1] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[2] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole Von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters*, 6(12):2326–2331, 2015.

[3] Daniel C. Elton, Zois Boukouvalas, Mark S. Butrico, Mark D. Fuge, and Peter W. Chung. Applying machine learning techniques to predict the properties of energetic materials. *Scientific Reports*, 8(1), jun 2018.

[4] B. C. Barnes, D. C. Elton, Z. Boukouvalas, D. E. Taylor, W. D. Mattson, M. D. Fuge, and P.W. Chung. Machine learning of energetic material properties. In *Proceedings of the 16th International Detonation Symposium, Cambridge, Maryland, USA*, 2018.

[5] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[6] James Barker, Johannes Bulin, Jan Hamaekers, and Sonja Mathias. Localized Coulomb descriptors for the Gaussian approximation potential. *arXiv preprint arXiv:1611.05126*, 2016.

[7] Felix A Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S Schoenholz, George E Dahl, Oriol Vinyals, Steven Kearnes, Patrick F Riley, and O Anatole von Lilienfeld. Machine learning prediction errors better than dft accuracy. *arXiv preprint arXiv:1702.05532*, 2017.

[8] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[9] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[10] David L Hall and James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[11] Federico Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013, 2013.

[12] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[13] T. Adalı, M. Anderson, and Geng-Shen Fu. Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging. *Signal Processing Magazine, IEEE*, 31(3):18–33, May 2014.

[14] T. Kim, H. T. Attias, S. Lee, and T. Lee. Blind source separation exploiting higher-order frequency dependencies. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):70–79, Jan 2007.

[15] Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[16] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

[17] Tapani Ristaniemi and Jyrki Joutsensalo. On the performance of blind source separation in cdma downlink. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 437–441, 1999.

[18] Andrew D Back and Andreas S Weigend. A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems*, 8(04):473–484, 1997.

[19] Intae Lee, Taesu Kim, and Te-Won Lee. Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87(8):1859–1871, 2007.

[20] Z. Boukouvalas. Development of ICA and IVA Algorithms with Application to Medical Image Analysis. *ArXiv e-prints*, January 2018.

[21] L. C. Blum and J.-L. Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.

[22] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9:513–530, 2018.

[23] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of Chemical Information and Modeling*, 56(10):1936–1949, 2016.

[24] Daniel C. Elton. Molecular Machine Learning Toolkit. `https://github.com/delton137/mmltoolkit`.

[25] Greg Landrum. RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

[26] Schrödinger Research, Inc. Machine learning & QSPR for materials. `https://www.schrodinger.com/science-articles/machine-learning-qspr-materials`, 2018.

[27] Lowell H. Hall and Lemont B. Kier. Electrotopological state indices for atom types: A novel combination of electronic, topological, and valence state information. *Journal of Chemical Information and Computer Sciences*, 35(6):1039–1045, 1995.

[28] Peter Ertl, Bernhard Rohde, and Paul Selzer. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *Journal of Medicinal Chemistry*, 43(20):3714–3717, oct 2000.

[29] X. Li and T. Adali. Independent component analysis by entropy bound minimization. *IEEE Transactions on Signal Processing*, 58(10):5151–5164, Oct 2010.

[30] Austin Kim, Zois Boukouvalas, and Daniel C. Elton. PyIVA. `https://github.com/zoisboukouvalas/pyiva`.

[31] MLSP-Lab. iva_laplace.m. `http://mlsp.umbc.edu/resources.html`.

[32] Zois Boukouvalas, Yuri Levin-Schwartz, Vince D Calhoun, and Tülay Adalı. Sparsity and independence: Balancing two objectives in optimization for source separation with application to fmri analysis. *Journal of the Franklin Institute*, 355(4):1873–1887, 2018.

[33] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[34] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, March 2011.

[35] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.