Finding Effective Ways to (Machine) Learn fMRI-based Classifiers from Multi-Site Data ⋆

Roberto Vega[1] and Russ Greiner[1]

[1] University of Alberta, Edmonton AB T6G 2R3, Canada
{rvega, rgreiner}@ualberta.ca

Abstract. Machine learning techniques often require many training instances to find useful patterns, especially when the signal is subtle in high-dimensional data. This is especially true when seeking classifiers of psychiatric disorders, from fMRI (functional magnetic resonance imaging) data. Given the relatively small number of instances available at any single site, many projects try to use data from multiple sites. However, forming a dataset by simply concatenating the data from the various sites, often fails, due to batch effects – that is, the accuracy of a classifier learned from such a multi-site datasets, is often worse than of a classifier learned from a single site. We show why several simple, commonly used, techniques – such as including the site as a covariate, z-score normalization, or whitening – are useful only in very restrictive cases. Additionally, we propose an evaluation methodology to measure the impact of batch effects in classification studies and propose a technique for solving batch effects under the assumption that they are caused by a linear transformation. We empirically show that this approach consistently improve the performance of classifiers in multi-site scenarios, and presents more stability than the other approaches analyzed.

Keywords: multi-site fMRI · batch effects · machine learning.

1 Introduction

Over the last years, many researchers have been seeking tools that can help with the diagnosis and prognosis of mental health problems. Research groups have used machine learning approaches in the analysis of fMRI data in order to build predictors that can diagnose, for example, attention deficit and hyperactivity disorders, mild cognitive impairment and Alzheimer's disease, schizophrenia, or autism [2]. The reported accuracy of the different tasks varies from chance level to $> 85\%$, depending on the task, dataset, features, and learning algorithm used for creating the classifier.

One of the main obstacles that limits the usability and generalization capabilities (to new instances) of machine learning approaches is the usually small

⋆ Supported by the Mexican National Council of Science and Technology (CONACYT), Canada's Natural Science and Engineering Research Council (NSERC) and the Alberta Machine Intelligence Institute (AMII).

number of instances (n) of the datasets used to train the models [2]. This is especially problematic when there are a large number of features (p), which might range from a few hundreds to millions depending on the approach, known as "*small n, large p*" [8]. This situation is undesirable because machine learning approaches assume that the training sample is a good approximation of the real distribution of the data, which might not be the case with only a few instances in a high dimensional space.

1.1 Multi-site data and batch effects

In order to mitigate this problem, many researchers use a larger datasets, formed by aggregating fMRI scans obtained at different locations into a single dataset. Unfortunately, inter-scanner variability, possibly caused by field strength of the magnet, manufacturer and parameters of the MRI scanner or radio-frequency noise environments [7], creates a second problem known as *batch effects* [12], which is technical noise that might confound the real biological signal. The main consequence of batch effects in prediction studies is that researchers have observed a *decrease* in classification accuracy on multi-site studies compared with that obtained using a single site [12, 16, 3].

An underlying assumption of machine learning is that the training set and test set are sampled from the same probability distribution. Because of batch effects, data coming from different sites follow different probability distributions, which might cause the predictors to have a decrease in performance. These discrepancies between the training and test sets are known as dataset shift [14]. This paper focuses on a specific subcase: Let $P_A(\,X, Y\,)$ be the joint distribution of the covariates X (the features extracted from the fMRI data) and the label Y (*e.g.*, healthy control or schizophrenia) of scanning site A, and $P_B(\,X, Y\,)$ be the corresponding probability distribution for a scanning site B, then $P_A(\,Y \,|\, X\,) \neq P_B(\,Y \,|\, X\,)$, and $P_A(\,X\,) \neq P_B(\,X\,)$, but there is a function $g(X)$ such that $P_A(\,Y \,|\, X\,) = P_B(\,Y \,|\, g(X)\,)$, and $P_A(\,X\,) = P_B(\,g(X)\,)$. This concept is exemplified in Figure 1.

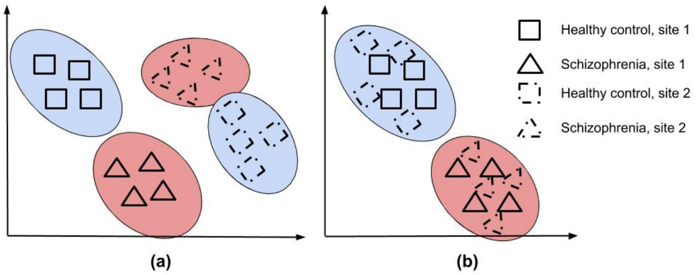

Fig. 1: (a) The dataset sampled from the scanning site 1 follows a different probability distribution than data scanned on site 2. (b) After applying $g(X)$ to the data of site 1 both sites follow the same probability distribution.

The problem of removing batch effects is closely related to that of domain adaptation in the computer vision community [5]. Although some of these approaches have been tested on fMRI data, the performance of classifiers learned from multi-site datasets is in many cases lower than using a single site [16]. The objective of this paper is to analyze some techniques for removing batch effects and the situations where they can be effectively used.

2 Machine learning and functional connectivity graphs

The standard approach for applying machine learning to fMRI data begins by parcellating the (properly preprocessed) brain volumes into m regions of interest. It then forms a symmetric $m \times m$ pairwise connectivity matrix, whose (i, j) entry each correspond to some measure of statistical dependence between regions i and j, whose upper-triangle is vectorized into a vector of length $p = \frac{1}{2}m(m-1)$.

The vectors corresponding to each of the n subjects in the training set are arranged into a matrix X of dimensions $n \times p$. Similarly, a vector Y of length n, contains the labels of X. Finally, this labeled training data $(X,\ Y)$ is given to a learning algorithm, that produces the final classifier. A detailed description of this procedure can be found elsewhere [15, 16].

A critical aspect in assessing the impact of batch effects in classification studies, as well as the effectiveness of the techniques applied to removed them, is the methodology used to measure the performance of the classifiers. Some studies pool together the data from the different sites and then randomly split the data into a training and test set, while others use the data from $(r-1)$ sites for training and the r^{th} site for testing [6, 11, 12, 1]. The first approach might mask the influence of batch effects because it artificially makes the distribution of the training set and test set more similar. This is an unrealistic scenario. In a real application, a clinician cares about the performance of the classifier on the patients that s/he is evaluating. The second approach is more realistic, but also more complicated. If there is indeed a function $g(X)$ that makes $P_1(\,Y \mid X\,) = P_2(\,Y \mid g(X)\,)$ then we need information from both scanning sites to learn it.

Therefore, we propose a third evaluation scenario: Fix the test set to be a specific subset of the data from site A. Then consider two training sets: just the remaining instances from site-A versus those remaining site-A instances *and also the instances from site B.* This approach, illustrated in Figure 2, has the advantage of identifying if there is a benefit of mixing data from different sites, or if it is better to train one classifier independently for every site. Note that this methodology requires having a labeled dataset from both scanning sites.

3 Batch effects correction techniques

3.1 Adding site as covariate

This technique involves augmenting each instance with its site information – encoded as a 1-hot-encoding. (That is, using r additional bit features, where the

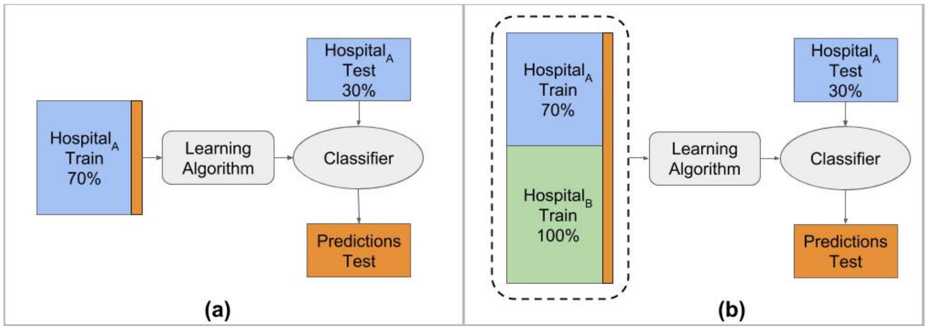

Fig. 2: Evaluating a classifier in single site (a) and multi-site (b) scenarios.

j^{th} feature is 1 if that instance comes from the j^{th} site, and the other features here are 0.)

When using a linear classifier, this method assumes that the only difference between sites is in the threshold that we use to classify an instance as belonging to one class, or another. If we assume that the decision function for one site is given by $w^T x = 0$, where the x vector represent the features and w is the vector of the coefficients (or weights) of the features, then the decision function for a second site is given by $w^T x + c = 0$. This method is effective when the batch effect is caused by a translation (adding a constant) to each instance of the dataset, but it will be ineffective otherwise. Figure 3(a) shows an illustration of this case. Note how the learned decision boundary is appropriate for one of the sites (red), but suboptimal for the other (blue). Note that this technique forces both decision boundaries to have the same slope, and only the bias changes.

3.2 Z-score normalization

This approach modifies the probability distribution of the features extracted from *both* sites, A and B, by making the values of each individual feature, for each site, zero-mean with unit variance – i.e., for each site, for the i^{th} feature, subject its mean (for that site), and divide by its empirical standard deviation (for that site). Using this technique, only the marginals are the same in both sites, but the covariance structure is not. Applying this "Z-score normalization" to the data from every scanning site independently, will effectively remove batch effects caused by translation and scaling of features (see Appendix A.1). However it fails with more complex transformations, such as rotations or linear transformations in general; see Figure 3(b). Note that this scaling and translation is in the feature space, and so it is different to the affine transformations that are corrected during the preprocessing stage (which are applied in the coordinate space).

3.3 Whitening

Whitening is a linear transformation that can be viewed as a generalization of the z-score normalization. Besides making the mean of every feature equal

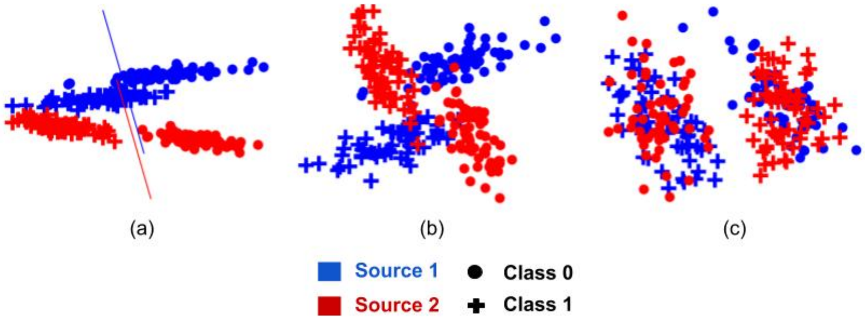

Fig. 3: Examples of linear transformation where the methods fail. (a) Including site as covariate, (b) z-score normalization, (c) whitening.

to zero and its variance equal to one, it also removes the correlation between features by making the overall covariance matrix the identity matrix. One of the most common procedures to perform this process is *PCA Whitening* [10]. This transformation first rotates the data, in each site, by projecting it into its principal components, and then it scales the rotated data by the square root of its eigenvalues (which represent the variance of each new variable in the PCA space). Applying this whitening transformation to every dataset independently will remove the batch effects caused by a rotation and translation of the datasets, since in this cases the principal components of the different sites will be aligned; see Appendix A.2 for the mathematical derivation. However, since there is no guarantee that the principal components will be aligned in general, it might not work with other linear transformations; see Figure 3(c).

3.4 Solving linear transformations

Note that z-score normalization and whitening solve specific cases of

$$X_B \quad = \quad \alpha X_A + \beta \quad \alpha \in \mathbb{R}^{p \times p}, \quad \beta \in \mathbb{R}^p \tag{1}$$

(corresponding to Equation 5 in Appendix A.2.) Z-score solves batch effects when the associated matrix α is diagonal, while whitening solves them when α is orthogonal with determinant 1. Nevertheless, both methods fail to solve batch effects for a general matrix α. Note also that the previous approaches did not explicitly compute α and β, but instead, applied a transformation that removed their effects under the specified circumstances. Of course, if we could compute α and β, or even a good approximation $\hat{\alpha}$ and $\hat{\beta}$, we could then solve for any batch effect corresponding to an arbitrary linear transformation.

For any two random vectors X_A and X_B, such that $X_B = \alpha X_A + \beta$:

$$\begin{array}{rclclclcl} \mu_B & = & E[X_B] & = & E[\alpha X_A + \beta] & = & \alpha E[X_A] + \beta & = & \alpha \mu_A + \beta \\ \Sigma_B & = & COV[X_B] & = & COV[\alpha X_A + \beta] & = & \alpha COV[X_A] \alpha^T & = & \alpha \Sigma_A \alpha^T \end{array} \tag{2}$$

Although we can obtain empirical estimates of $\mu_A,\ \mu_B,\ \Sigma_A,\ \Sigma_B$ from the dataset, the problem is in general ill-defined – *i.e.*, there is an infinite number of solutions. Now note that every site includes (at least) two different subpopulations – *e.g.*, healthy controls versus cases (perhaps people with schizophrenia). Each subpopulation has its own mean vector and covariance matrix (μ_A^{HC}, μ_A^{SCZ}, μ_B^{HC}, μ_B^{SCZ}, and $\Sigma_A^{HC}, \Sigma_A^{SCZ}, \Sigma_B^{HC}, \Sigma_B^{SCZ}$) . A reasonable assumption is that the batch effects affect both populations in the same way, but by computing the mean and covariance matrix of every population and site independently we are effectively increasing the number of equations available. We can then get an estimate for α and β as follows:

$$\hat{\alpha},\ \hat{\beta} \quad = \quad \underset{\alpha,\beta}{\arg\min} \sum_{j \in \{HC,SCZ\}} \sqrt{p}||\mu_B^j - (\alpha\mu_A^j + \beta)||_2 + ||\Sigma_B^j - (\alpha\Sigma_A^j\alpha^T)||_F \quad (3)$$

where p is the dimensionality of the feature set, and $||\cdot||_F$ is the Frobenius norm of a matrix. Note that it is possible to combine data from more than two datasets by finding a linear transformation for every pair of sites.

4 Experiments and Results

4.1 Dataset

We applied the four aforementioned methods to the task of classifying healthy controls and people with schizophrenia using the data corresponding to the Auditory Oddball task to the FBIRN phase II dataset, which is a multisite study developed by the Function Biomedical Informatics Research Network (FBIRN). Keator *et al.* provides a complete description of the study [9].

After preprocessing the data, we eliminated the subjects who presented head movement greater than the size of one voxel at any point in time in any of the axis, a rotation displacement greater than 0.06 radians, or that did not pass a visual quality control assessment. The original released data contains scans extracted from 6 different scanning sites; however, we only used 4 of them. One of the sites was discarded because it lacked T1-weighted images, which were required as part of our preprocessing pipeline. The second discarded site contained only 6 subjects (5 with schizophrenia) after the quality control assessment, so it was not suitable for our analysis. In summary, we have 21 participants from Site 1, 22 from Site 2, 23 from Site 3 and 23 from Site 4. In all cases, the proportion of healthy controls vs people with schizophrenia is $\sim 50\%$.

4.2 Experiments and results

To obtain the feature vector of every fMRI scan, we used the subset corresponding to the Fronto-Parietal Network for a total of $k = 25$ out of the 264 regions of interest defined by Power *et al.* [13]. The time series corresponding to every region was simply the average time series of all the voxels inside the region. In

Table 1: Average accuracy after correcting batch effects. The number in entry (i, j) is the accuracy, over instances from the target site i, of the classifier learned by adding all of site j to the training subset of site i. The colored cells indicate results whose difference improves (green) or decrease (red) relative to the single site classification.

	S 1	S 2	S 3	S 4
S 1	62.8	72.3	65.7	67.3
S 2	67.8	66.4	70.0	59.5
S 3	55.0	60.9	58.3	56.9
S 4	62.3	57.8	76.4	71.4

(a) No correction

	S 1	S 2	S 3	S 4
S 1	62.8	70.7	64.7	68
S 2	67.1	66.4	68.1	57.6
S 3	55.7	57.6	58.3	56.9
S 4	67.1	57.6	75.7	71.4

(b) Site as covariate

	S 1	S 2	S 3	S 4
S 1	62.8	64.7	57.6	65.7
S 2	68.5	66.4	67.6	62.3
S 3	48.0	54.0	58.3	58.0
S 4	63.5	56.0	74.0	71.4

(c) Z-score normalization

	S 1	S 2	S 3	S 4
S 1	62.8	55.7	52.8	49.5
S 2	51.6	66.4	52.1	50.4
S 3	54.2	54.2	58.3	53.8
S 4	50.7	47.3	52.6	71.4

(d) Whitening

	S 1	S 2	S 3	S 4
S 1	62.8	65.9	66.4	66.2
S 2	66.6	66.4	67.8	67.8
S 3	49.5	50.2	58.3	51.4
S 4	73.5	72.8	73.5	71.4

(e) Linear transformation

order to obtain the functional connectivity matrix, we computed the Pearson's correlation between the time series of all $\binom{k}{2}$ pairs of regions.

We produced classifiers using a support vector machine (SVM) with linear kernel using the SVMLIB library [4]. The parameters of the SVM were set using cross validation. We applied the batch effect correction techniques previous to merging the datasets into a single training set, and repeated the experiment 15 times with different train/test splits. All the parameters required for the batch effects correction techniques were obtained using only the training sets. Table 1 reports the average accuracy over the 15 rounds.

5 Discussion

In each of the sub-tables in Table 1, the (i, j) entry represent the average accuracy when the training set has instances from the ith and jth site, and the test set has instances only from the ith site. Ideally, all the off-diagonal values should be higher than the diagonal ones; however, this is not the case. In most of the cases we have mixed and inconsistent results. The only method that consistently improves the performance of the classifiers is the one that solves for arbitrary linear transformations (Table 1e). Note that site S3 is an exception, where we do not see any improvement; however, this particular site has a low performance even in the single site scenario. It is likely that the signal in this particular site is too low and cannot be properly detected by the used methods.

These results reinforce the idea that batch effects play predominant role in classification studies, and motivate the need to develop techniques that address them in order to be able to effectively combine multi-site datasets. We can additionally conclude that whitening, z-score normalization and adding the site as covariate are insufficient to solve batch effects in fMRI data. Our method for solving linear transformations is the one who consistently improves the results in a multi-site scenario, indicating that it is a step in the right direction.

References

1. Abraham, A., Milham, M.P., Di Martino, A., Craddock, R.C., Samaras, D., Thirion, B., Varoquaux, G.: Deriving reproducible biomarkers from multi-site resting-state data: An autism-based example. NeuroImage **147**, 736–745 (2017)
2. Arbabshirani, M.R., Plis, S., Sui, J., Calhoun, V.D.: Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. Neuroimage **145**, 137 – 165 (2016)
3. Brown, M.R.G., Sidhu, G.S., Greiner, R., Asgarian, N., Bastani, M., Silverstone, P.H., Greenshaw, A.J., Dursun, S.M.: ADHD-200 global competition: diagnosing ADHD using personal characteristic data can outperform resting state fMRI measurements. Frontiers In Systems Neuroscience **6**, 69 (2012)
4. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011)
5. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374 (2017)
6. Gheiratmand, M., Rish, I., Cecchi, G.A., Brown, M.R.G., Greiner, R., Polosecki, P.I., Bashivan, P., Greenshaw, A.J., Ramasubbu, R., Dursun, S.M.: Learning stable and predictive network-based patterns of schizophrenia and its clinical symptoms. NPJ Schizophrenia **3**, 22 (2017)
7. Greve, D.N., Brown, G.G., Mueller, B.A., Glover, G., Liu, T.T.: A survey of the sources of noise in fMRI. Psychometrika **78**(3), 396 – 416 (2013)
8. Hastie, T.J., Tibshirani, R.J., Friedman, J.H.: The elements of statistical learning : data mining, inference, and prediction. Springer series in statistics, Springer, New York (2009)
9. Keator, D.B., et al.: The function biomedical informatics research network data repository. NeuroImage **124, Part B**, 1074 – 1079 (2016), sharing the wealth: Brain Imaging Repositories in 2015
10. Kessy, A., Lewin, A., Strimmer, K.: Optimal whitening and decorrelation. (2015)
11. Nielsen, J.A., Zielinski, B.A., Fletcher, P.T., Alexander, A.L., Lange, N., Bigler, E.D., Lainhart, J.E., Anderson, J.S.: Multisite functional connectivity mri classification of autism: Abide results. Frontiers in human neuroscience **7**, 599 (2013)
12. Olivetti, E., Greiner, S., Avesani, P.: ADHD diagnosis from multiple data sources with batch effects. Frontiers In Systems Neuroscience **6**, 70 (2012)
13. Power, J.D., Cohen, A.L., Nelson, S.M., Wig, G.S., Barnes, K.A., Church, J.A., Vogel, A.C., Laumann, T.O., Miezin, F.M., Schlaggar, B.L., Petersen, S.E.: Functional network organization of the human brain. Neuron **72**(4), 665 – 678 (2011)
14. Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: When Training and Test Sets Are Different: Characterizing Learning Transfer. (2012)
15. Richiardi, J., Achard, S., Bunke, H., Van De Ville, D.: Machine learning with brain graphs: Predictive modeling approaches for functional imaging in systems neuroscience. IEEE Signal Processing Magazine **30**(3), 58–70 (2013)
16. Vega Romero, R.I.: The challenge of applying machine learning techniques to diagnose schizophrenia using multi-site fMRI data. [2017] (2017)

A Mathematical derivations

A.1 z-score normalization

Let X_i^A and X_i^B represent the values of the i^{th} feature extracted from scanning sites A and B respectively. Then, we can represent the operations of scaling and translation as:

$$X_i^B \quad = \quad \alpha_i X_i^A + \beta_i, \qquad i = 1, 2, \ldots, m \tag{4}$$

where α_i and β_i are the scaling and translation coefficients of the ith feature. In order to apply Z-score normalization we need to subtract the mean of every feature and divide by the standard deviation. Then the z-score normalized features from scanning sites A and B, $\bar{X_i^A}$, $\bar{X_i^B}$, are:

$$\begin{aligned}
\bar{X_i^A} &= \frac{X_i^A - E[X_i^A]}{\sqrt{Var(X_i^A)}} \\
\bar{X_i^B} &= \frac{X_i^B - E[X_i^B]}{\sqrt{Var(X_i^B)}} \\
&= \frac{\alpha_i X_i^A + \beta_i - E[\alpha_i X_i^A + \beta_i]}{\sqrt{Var(\alpha_i X_i^A + \beta_i)}} \\
&= \frac{\alpha_i \left(X_i^A - E[X_i^A] \right)}{\sqrt{\alpha_i^2 Var(X_i^A)}} \\
&= \frac{X_i^A - E[X_i^A]}{\sqrt{Var(X_i^A)}}, \quad \text{for } \alpha_i > 0 \\
&= \bar{X_i^A}
\end{aligned}$$

Therefore, after applying Z-score normalization, we are effectively removing the effects of translation and scaling.

A.2 Whitening

To see why whitening removes the effects of rotation and scaling, consider the case where the datasets X_B is a rotation and translation of X_A. This can be represented in matrix form as:

$$X^B \quad = \quad X^A \alpha + \mathbf{1}\beta^T \quad \alpha \in \mathbb{R}^{p \times p}, \;\; \beta \in \mathbb{R}^p \tag{5}$$

where α is a rotation matrix – *i.e.*, is an orthogonal matrix with determinant $\det(\alpha) = 1$. The zero-mean datasets, $\bar{X}_A$, can be obtained as:

$$\begin{aligned}
\bar{X}_A &= X_A - \mathbf{1}E[X_A] \\
E[X_A] &= [\ E[X_A^1],\ E[X_A^2],\ \ldots,\ E[X_A^p]\ ]
\end{aligned} \tag{6}$$

while for the case of $\bar{X_B}$:

$$\begin{aligned}\bar{X}_B &= X_A\alpha + \mathbf{1}\beta^T - \mathbf{1}E[X_A\alpha + \mathbf{1}\beta^T] \\ &= X_A\alpha + \mathbf{1}\beta^T - \mathbf{1}\left(E[X_A\alpha] - E[\mathbf{1}\beta^T]\right) \\ &= \left(X_A - \mathbf{1}[X_A]\right)\alpha \\ &= \bar{X}_A\alpha\end{aligned} \tag{7}$$

The eigenvalues of the covariance matrix $\Sigma_A = \frac{1}{n-1}\bar{X}_A^T\bar{X}_A$ are obtained by solving the equation $\det(\Sigma_A - \lambda I) = 0$. For the special case when α is a rotation matrix[1], $\alpha^T = \alpha^{-1}$, the eigenvalues of the covariance matrix of $\bar{X}_B$:

$$\begin{aligned}0 &= \det\left(\frac{1}{n-1}(\bar{X}_A\alpha)^T(\bar{X}_A\alpha) - \lambda I\right) \\ &= \det\left(\frac{1}{n-1}\alpha^T\bar{X}_A^T\bar{X}_A\alpha - \lambda I\right) \\ &= \det\left(\alpha^T\Sigma_A\alpha - \lambda I\right) \\ &= \det\left(\alpha^{-1}\Sigma_A\alpha - \alpha^{-1}\lambda I\alpha\right) \\ &= \det\left(\alpha^{-1}(\Sigma_A - \lambda I)\alpha\right) \\ &= \det(\alpha^{-1})\det(\Sigma_A - \lambda I)\det(\alpha) \\ &= \det(\Sigma_A - \lambda I)\end{aligned} \tag{8}$$

As for the eigenvectors: if v is an eigenvector of Σ_A with an associated eigenvalue λ, then $\Sigma_A v = \lambda v$. Doing some mathematical manipulations:

$$\begin{aligned}\alpha\Sigma_A v &= \alpha\lambda v \\ \alpha\Sigma_A I v &= \alpha\lambda v \\ \alpha\Sigma_A\alpha^{-1}\alpha v &= \lambda\alpha v \\ \Sigma_B(\alpha v) &= \lambda(\alpha v)\end{aligned} \tag{9}$$

Equations 8 and 9 show that, when the transformation matrix α is an orthogonal matrix with positive determinant, X_A and X_B will have the same eigenvalues, and the eigenvectors of X_B are just a rotation of the eigenvectors of X_A. Therefore, by projecting the data into those eigenvector, we obtain the exact same representation, removing the effects of translation and rotation.

[1] All orthogonal matrices α have a determinant equal to +1, or -1. If it is positive, α is a rotation matrix. When the determinant is negative, it is a reflection matrix.