Alternating Minimization for Regression Problems with Vector-valued Outputs[Mixed]

Nakula Neeraje (190525)
Electrical Engineering
IIT Kanpur
Kanpur, India
nakula@iitk.ac.in

Sarvesh Chandra (190773)
Electrical Engineering
IIT Kanpur
Kanpur, India
csarvesh@iitk.ac.in

***Abstract*—For regression problems dealing with vector-valued outputs, there often exists a noise correlation across the multiple tasks. Hence, it is seen that the maximum likelihood estimate(MLE) which takes the noise correlation structure into account does the best job in estimation. It is much better than Ordinary least squares(OLS) which in contrary does not take the noise correlation into consideration. Unfortunately, solving the MLE is a non-convex problem and is hence not efficiently solvable. We show that the output of alternating minimization procedure is very close to the result of the MLE, while also being more efficient to compute.**

I. Introduction

Regression problems arise when we want to predict an output, given a set of input variables. Vector-valued regression problems, in specific are when we want to predict multiple outputs[1] when given a set of predictor variables. It is generally seen that there is a correlation in the noise or the error terms across different tasks in multi-task learning.
In such a case, the MLE would be a much better estimator when compared to OLS. This is because the MLE takes the noise into account when calculating the likelihood estimate. An exception is when the noise is uncorrelated across tasks. Regardless, even with correlated noise, solving the MLE is a non-convex optimization problem which makes it less efficient in finding the optimal solution.
Hence, in this report, we would be exploring the alternating minimization approach to solving the optimization problem and comparing it with the naive OLS and the non-convex MLE and testing it for the accuracy of results by checking the error bounds. The Alt-Min procedure, in an alternating fashion, estimates the regression parameters and the noise covariance matrix. The individual problems are convex, and in fact have analytical solutions. Regardless, the general problem is non-convex. Thus, there is no guarantee of global convergence.
The model we consider is the Pooled Model for parameter estimation.

II. Pooled Model

In a pooled model, as the name suggests, a single co-efficient matrix is used across all tasks and data samples. Although rather restrictive, the pooled model has multiple applications and can be thought of as a generalization of the Multiple Regression Model.

$\mathcal{D} = \{(X_i, y_i), ..., (X_n, y_n)\}$. Here, the $i-$th data type X_i is of the form $X_i \in \mathbf{R}^{m\times d}$, and the output y_i is of the form $y_i \in \mathbf{R}^m$.
d is the dimensionality of the samples while m stands for the number of tasks. The motive is to learn the co-efficient weights w$\in \mathbf{R}^d$ such that $y \approx X$w, for any new data sample X and its corresponding output y. We assume that the data is generated in the following manner, being consistent with the pooled model:

$$y_i = X_i w_* + \eta_i, \quad 1 \leq i \leq n \tag{1}$$

Here, w_* is the optimal paramter vector that we want to learn. The data points X_i, $1 \leq i \leq n$ are identically and independently(i.i.d.) sampled from an underlying probability distribution. The noise vectors $\eta_i \sim \mathcal{N}(0, \Sigma_*)$ are i.i.d. sampled from a Gaussian with mean at origin and a covariance matrix Σ_*.

A naive way of estimating w$_*$ is to ignore the noise correlation and treat the problem as a generic multiple regression problem across all data samples. This is basically the Ordinary Least Squares(OLS) approach:

$$w_{OLS} = arg\ \min_{w} \frac{1}{n}\sum_{i=1}^{n} ||y_i - X_i w||_2^2. \tag{2}$$

We show that solving for w$_{OLS}$ is a convex optimization problem, and in fact has an analytical solution:

$$w_{OLS} = (\sum_{i=1}^{n} X_i^T X_i)^{-1} \sum_{i=1}^{n} X_i^T y_i \tag{3}$$

See appendix for a detailed proof of the above result.
We see that the result will become increasingly more accurate as $n \to \infty$. Hence, we compare the above result obtained with other algorithms like Alternating Minimization(Alt-Min) and Maximum likelihood estimate(MLE) for a fixed n.

To leverage the noise correlation and to devise an algorithm to estimate w$_*$ and Σ_*, we first look at the joint maximum likelihood estimation problem(MLE) for (w, Σ).
We take the log likelihood, and remove the constant terms [2]:

Algorithm 1 Alternating Minimization for the Pooled Model

Require: $D = \{(X_1, y_1).....(X_{2nT}.y_{2nT})\}$ ▷ 2nT Data samples
1: Partition $D = \{D_0^{\Sigma}, D_0^{w}, ..., D_T^{\Sigma}, D_T^{w}\}$ ▷ Each box in the partition contains n data samples
2: Initialize $w_0 = 0$
3: **for** $t = 0, ..., T-1$ **do** ▷ We run T iterations
4: Noise Covariance Estimation: $\Sigma_t = \frac{1}{n}\sum_{i \in D_t^{\Sigma}} (y_i - X_i w_t)(y_i - X_i w_t)^T$
5: Parameter Estimation: $w_{t+1} = arg\ min_w \frac{1}{n}\sum_{i \in D_t^{w}} ||\Sigma_t^{-\frac{1}{2}}(y_i - X_i w)||_2^2$
6: **end for**
7: **return** w_T ▷ The parameter vector estimated is w_T

$$arg \max_{w,\Sigma} -\log|\Sigma| - \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i w)^T \Sigma^{-1}(y_i - X_i w). \quad (4)$$

Jointly optimizing for $(\hat{w}, \hat{\Sigma})$ to find the absolute maxima is a non-convex optimization problem. Hence, we now look at the Alternating Minimization approach where we try to minimize the MLE with respect to w_* and Σ_* individually, while keeping the other variable fixed. These individual problems are convex, and even have analytical solutions. The solutions to the individual problems are:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i w)(y_i - X_i w)^T \quad (5)$$

$$w = arg\ min_w \frac{1}{n}\sum_{i=1}^{n} ||\Sigma^{-\frac{1}{2}}(y_i - X_i w)||_2^2 \quad (6)$$

It is easy to see that w has an analytical form because it can be converted to the Ordinary least Squares form just by a simple change in coefficients.
We prove the above results in the appendix.
On a side note, assuming Σ_* is know apriori, the MLE optimization problem would just be:

$$w_{MLE} = arg\ min_w \frac{1}{n}\sum_{i=1}^{n} ||\Sigma_*^{-\frac{1}{2}}(y_i - X_i w)||_2^2 \quad (7)$$

It is easy to see that this has an analytical solution as well because it is of the same form as the Ordinary Least Squares(OLS) problem.
We can exploit the cyclical structure by using the Alternating-Minimization algorithm [2]. This is because we can continually refine the predictions of the Noise Covariance Estimation and the Parameter Estimation in each iteration.
Given an initial w_0 and fresh data samples in every iteration for both the Noise Covariance estimation and the Parameter estimation, we use the analytical solution of the two convex problems to approximate the solution of the non-convex problem.
More importantly, it is not guaranteed that we reach the global optima in the Alternating Optimization procedure, because the original problem is non-convex. We most likely just reach a local optima.

A. Error Bounds

We use the error bounds of w as given in [2] for the Ordinary Least Squares estimate(OLS), the Maximum Likelihood estimate and the Alternating Maximization estimate.

$$\mathbb{E}_{\mathbf{X}}[||X(w_{OLS} - w_*)||_2^2] \leq \frac{Cd\log n}{n}.\frac{tr(\Sigma_*)}{m} \quad (8)$$

$$\mathbb{E}_{\mathbf{X}}[||X(w_{MLE} - w_*)||_2^2] \leq \frac{Cd\log n}{n}.\frac{m}{tr(\Sigma_*^{-1})} \quad (9)$$

$$\mathbb{E}_{\mathbf{X}}[||X(w_{AM} - w_*)||_2^2] \leq \frac{8Cd\log n}{n}.\frac{m}{tr(\Sigma_*^{-1})} + \epsilon \quad (10)$$

Here, we take C = $\frac{n}{(m+d)(\log(m+d))}$ and $\epsilon = \log 1/T$ where T is the number of iterations in Algorithm 1.
We see that the error bound is tighter for the Alt-Min algorithm when compared to the MLE algorithm. It was shown in [1] that:

$$\frac{m}{tr(\Sigma_*^{-1})} \leq \frac{tr(\Sigma_*)}{m} \quad (11)$$

The gap increases with increased correlation of the noise across tasks. Hence, MLE has a tighter bound when compared to OLS.
It is to be noted that we are sampling the rows of X independently for this analysis.

III. Experiments

We perform our experiments by sampling the noise from $N(0, \Sigma_*)$, where, for m=2,

$$\Sigma_* = \begin{pmatrix} 1 & 1-\epsilon \\ 1-\epsilon & 1 \end{pmatrix} \quad (12)$$

We generalize Σ_* for $m \geq 2$ as follows:

$$\Sigma_* = \begin{pmatrix} 1 & 1-\epsilon & \mathbf{0} \\ 1-\epsilon & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I_{m-2,m-2}} \end{pmatrix} \quad (13)$$

Here, we present the plots obtained by calculating AltMin (shown in green), OLS (shown in red) and MLE (shown in blue) estimates for 3 different distributions of data against d, epsilon, n and m.
The three different distributions are ($X_{m,d}$):
Case I: w_* is a random vector and

$$X_i \sim N(\mathbf{0_{d,1}}, \mathbf{I_{d,d}})\ \ 1 \leq i \leq m \quad (14)$$

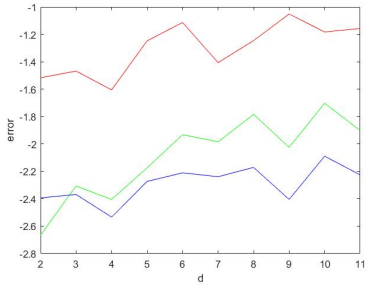

Fig. 1: Variation of Error with d. [m=2, n=50, $\epsilon = 0.005$]

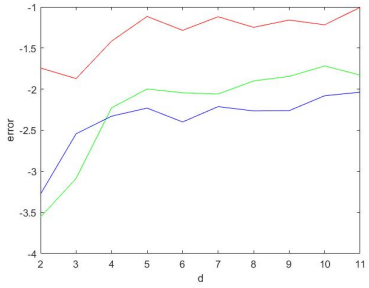

Fig. 2: Variation of Error with d with altered Data. [m=2, n=50, $\epsilon = 0.005$]

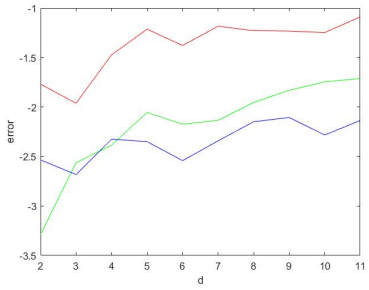

Fig. 3: Variation of Error with d with altered Data and optimal weights. [m=2, n=50, $\epsilon = 0.005$]

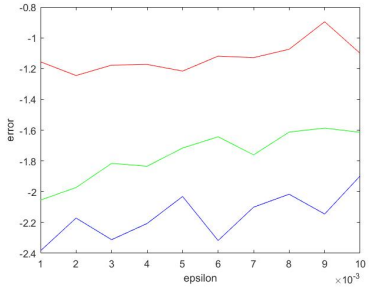

Fig. 4: Variation of Error with ϵ, [m=2, n=50, d=10]

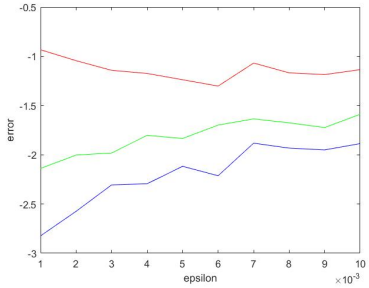

Fig. 5: Variation of Error with ϵ with altered Data, [m=2, n=50, d=10]

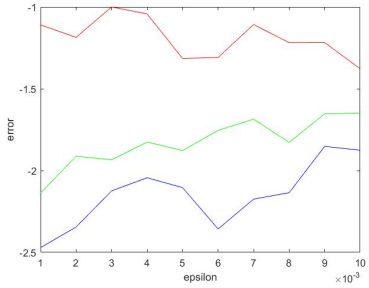

Fig. 6: Variation of Error with ϵ with altered Data and altered optimal weights, [m=2, n=50, d=10]

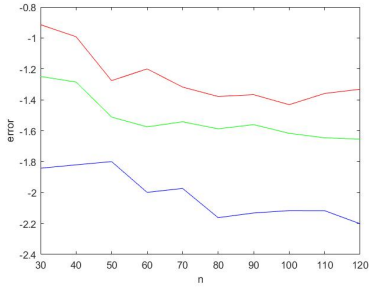

Fig. 7: Variation of Error with n, [m=5, d=20, $\epsilon = 0.005$]

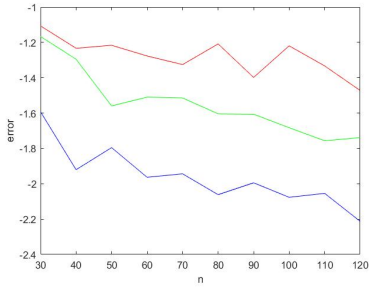

Fig. 8: Variation of Error with n with altered Data, [m=5, d=20, $\epsilon = 0.005$]

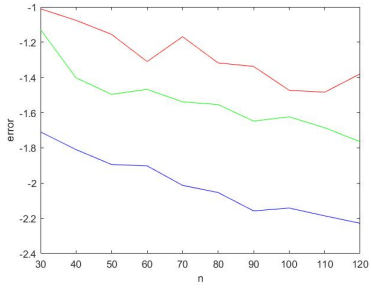

Fig. 9: Variation of Error with n with altered Data and altered optimal weights, [m=5, d=20, $\epsilon = 0.005$]

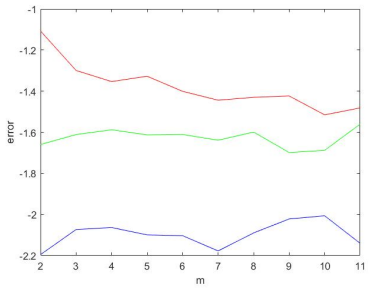

Fig. 10: Variation of Error with m [n=100, d=20, $\epsilon = 0.005$]

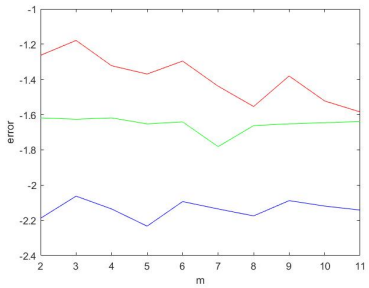

Fig. 11: Variation of Error with m with altered Data [n=100, d=20, $\epsilon = 0.005$]

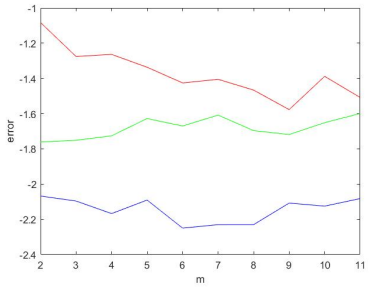

Fig. 12: Variation of Error with m with altered Data and altered optimal weights [n=100, d=20, $\epsilon = 0.005$]

Case II: w_* is a random vector and

$$X_i \sim N(\mathbf{10000_{d,1}}, \mathbf{I_{d,d}}) \ \ 1 \leq i \leq m \tag{15}$$

Case III: w_* is 100*$\mathbf{1_{d,1}}$ + 100*$\mathbf{random(0,1)_{d,1}}$

$$X_i \sim N(\mathbf{10000_{d,1}}, \mathbf{I_{d,d}}) \ \ 1 \leq i \leq m \tag{16}$$

We vary the parameters accordingly depending on the plot. The y-axis corresponds to the L2 norm of $(w - w_*)$. The y-axis is in log scale for better data representation. [1] uses the initial distribution as given in Case 1. We try to tinker with the original data so as to generate greater insight about the algorithms.

A. Varying the number of dimensions in the data

In Fig 1,2,3: We vary the number of dimensions with Fig 1 having the case I representation and so on. We observe the error for OLS is greater than the error for MLE and AltMin. We also notice that there is an upward trend in the error as the number of dimensions increase. This makes sense because we keep the number of data samples constant, even when increasing the number of data dimensions. We also observe that there is slight overlap between MLE and AltMin for low d, but MLE clearly performs better than AltMin as we increase the number of dimensions. We assume this is because local optima obtained in the AltMin algorithm strays away from the global maxima obtained in the MLE algorithm as we increase the number of dimensions.

B. Varying the value of ϵ in Σ_*

In Fig 4,5,6: We vary the value of ϵ with Fig 4 having the case I representation and so on. We again observe that the error for OLS is greater than the error for MLE and AltMin. We also observe that while the error remains mostly consistent for OLS when we vary ϵ (because the error for OLS does not depend on ϵ), the error for MLE and AltMin increase with increase in ϵ. The errors of MLE and AltMin also seems to converge to the error of OLS with increase in ϵ. This is because of the way we have defined Σ_*. As $\epsilon \to 1$, $\Sigma_* \to \mathbf{I}$. This would lead to the noise being uncorrelated and the MLE error would be identical to the OLS error. Hence, we see the convergence of errors when we increase ϵ.

C. Increasing the value of n in the data

In Fig 7, 8, 9: We vary the number of data samples with Fig 7 having the case I representation and so on. Once again, the error for OLS is greater than the error for MLE and AltMin. We notice that the error decreases for all 3 algorithms with increase in the number of data samples. This makes sense intuitively because we get better results with greater number of input samples. We can see that the error bounds get tighter in all three algorithms when n increases. As can be seen from the error bounds and the plots, the error terms for MLE decreases faster than Alt-Min, which in turn decreases faster than OLS.

D. Increasing the number of tasks in the data samples

In Fig 10, 11, 12: We vary the number of tasks in the data samples with Fig 10 having the case I representation and so on. We again observe that the error for OLS is greater than the error for MLE and AltMin. Interestingly, we also observe that the error for OLS increases with the increase in number of tasks while the error for MLE and Alt-Min remain mostly consistent with the increase. This is because OLS does not take the noise correlation across tasks in a data sample into account. Hence, increasing the number of tasks in a data sample is akin to increasing the number of data samples in the case of OLS. Because MLE and Alt-Min already take the Noise Correlation structure into account, we do not see any significant change in their errors.

The distributions as described in Case I, II and III are used to check the correctness of the algorithm for a real world scenario as well - where the data points and weights would be different by orders of magnitude from the noise. From the plots, it is clear that the error and trends in all three cases is practically the same for each varying parameter, thereby proving that the algorithm can accurately handle real world scenarios as well.

We also compute the error bounds (8) (9) (10) as specified in the paper. We also use the following relationship to simplify the calculation: For $H \in \{OLS, MLE, AltMin\}$, as $(w_H - w_*)$ is constant w.r.t. X,

$$\mathbb{E}_{\mathbf{X}}[||X(w_H - w_*)||_2^2] = (w_H - w_*)^T \mathbb{E}_{\mathbf{X}}[X^T X](w_H - w_*) \tag{17}$$

For each of OLS, MLE, AltMin, we take $C = \frac{n}{(m+d)(\log(m+d))}$ and for AltMin we take $\epsilon = \log 1/T$ where T is the number of iterations in Algorithm 1.

We verified that the error bounds are satisfied for each of OLS, MLE, AltMin in each run of the respective algorithms.

IV. Conclusion, Suggested Improvements and Future Works

In our analysis, we proved the correctness of the AltMin algorithm by theoretically deriving the update equations for both Σ and w as mentioned in [2]. We also experimentally proved that the error bound in case of all three algorithms holds across all the runs. In each run, AltMin, OLS and MLE all converged to w_*. We also see that AltMin somewhat follows the trend of MLE in each of the plots, which was the basic motivation behind the design of the algorithm. We can conclude that AltMin is clearly more precise than OLS and in some cases is almost as precise as MLE, though being significantly easier to compute practically than MLE. We have critically analysed the effect of each and every parameter on the precision of the three estimates.

We have gone beyond the suggested implementation in [2] by incorporating three different types of data distributions because the choice of X and η both being sampled from

spherical multivariate gaussian is not representative of practical scenarios where data points are sampled from distributions way different from the noise. Another criticism of the algorithm in [2] is that it does not take many iterations to converge to the optimal value of w and any further iterations only shift w from its optimal value. We have therefore improved the AltMin algorithm during the implementation by taking the minimum error encountered in the T iterations. A third criticism of the implementation in [2] is that the AltMin estimate for Σ is not very close to Σ_*. We improved upon the estimate by employing the method in Case II and Case III, which as described before, are more practical estimates than Case I suggested in [2].

We wish to improve upon the estimate of Σ_* from the implementation of the algorithm as well as analyse it in greater theoretical detail. If we can estimate Σ_* and w_* together reasonably, we would move closer to the global optimum of the joint MLE, which can be a thorough topic for future discussions.

V. Appendix

A. AltMin update equation for Σ

$$F(\Sigma, w) = -\log|\Sigma| - \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i w)^T\Sigma^{-1}(y_i - X_i w) \tag{18}$$

$$\frac{\partial F(\Sigma, w)}{\partial \Sigma} = -\frac{\partial \log|\Sigma|}{\partial \Sigma} - \frac{1}{n}\sum_{i=1}^{n}\frac{\partial (y_i - X_i w)^T\Sigma^{-1}(y_i - X_i w)}{\partial \Sigma} \tag{19}$$

Using,

$$\frac{\partial a^T X^{-1} b}{\partial X} = -X^{-T}ab^T X^{-T} \tag{20}$$

$$\frac{\partial F(\Sigma, w)}{\partial \Sigma} = -\frac{1}{|\Sigma|}\frac{\partial|\Sigma|}{\partial \Sigma} + \frac{1}{n}\sum_{i=1}^{n}\Sigma^{-T}(y_i - X_i w)(y_i - X_i w)^T\Sigma^{-T} \tag{21}$$

Using [3],

$$\frac{\partial|\Sigma|}{\partial \Sigma} = |\Sigma|\Sigma^{-T} \tag{22}$$

$$\frac{\partial F(\Sigma, w)}{\partial \Sigma} = -\Sigma^{-T} + \frac{1}{n}\sum_{i=1}^{n}\Sigma^{-T}(y_i - X_i w)(y_i - X_i w)^T\Sigma^{-T} \tag{23}$$

In AltMin, we need to maximize $F(\Sigma, w)$ w.r.t Σ while computing optimal Σ.

Thus,

$$\frac{\partial F(\Sigma, w)}{\partial \Sigma} = \mathbf{O} \tag{24}$$

$$-\hat{\Sigma}^{-T} + \frac{1}{n}\sum_{i=1}^{n}\hat{\Sigma}^{-T}(y_i - X_i w)(y_i - X_i w)^T\hat{\Sigma}^{-T} = \mathbf{O} \tag{25}$$

$$\hat{\Sigma}^T = \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i w)(y_i - X_i w)^T \tag{26}$$

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i w)(y_i - X_i w)^T \tag{27}$$

B. Generalized solution for Ordinary least squares problem

$$F(w) = \sum_{i \in D_t^w} ||A_i w - B_i||_2^2 \tag{28}$$

$$min_w F(w) \tag{29}$$

$$\frac{\partial F(w)}{\partial w} = \frac{\partial}{\partial w}\sum_{i \in D_t^w}(A_i w - B_i)^T(A_i w - B_i) \tag{30}$$

$$= \frac{\partial}{\partial w}\sum_{i \in D_t^w}(w^T A_i^T A_i w + B_i^T B_i - w^T A_i^T B_i - B_i^T A_i w) \tag{31}$$

$$= \frac{\partial}{\partial w}\sum_{i \in D_t^w}(2A_i^T A_i w - 2A_i^T B_i) = \mathbf{O} \tag{32}$$

Hence:

$$w_{OLS} = (\sum_{i \in D_t^w} A_i^T A_i)^{-1}\sum_{i \in D_t^w} A_i^T B_i \tag{33}$$

C. Analytical solution of MLE

It can be seen that the analytical solution of MLE(when we assume Σ_* to be given apriori) is just the special case of the OLS problems where:

$$A_i = \Sigma_*^{-\frac{1}{2}} X_i \tag{34}$$

$$B_i = \Sigma_*^{-\frac{1}{2}} y_i \tag{35}$$

D. AltMin update equation for w

Similarly, the analytical solution for the minimization of w_{t+1} in Algorithm 1 is a special case of the OLS problems where:

$$A_i = \Sigma_t^{-\frac{1}{2}} X_i \tag{36}$$

$$B_i = \Sigma_t^{-\frac{1}{2}} y_i \tag{37}$$

E. Summation of PSD matrices is PSD

Let A_i, $1 \leq i \leq n$ be PSD matrices. Hence, for any given vector u, $u^T A_i u > 0$, $1 \leq i \leq n$.

We see that, $u^T \sum_{i=1}^{n} A_i u > 0$ as well. This means that $\sum_{i=1}^{n} A_i$ is PSD.

REFERENCES

[1] P. Rai, A. Kumar, and H. Daume, "Simultaneously leveraging output and task structures for multiple-output regression," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/4dcae38ee11d3a6606cc6cd636a3628b-Paper.pdf

[2] P. Jain and A. Tewari, "Alternating minimization for regression problems with vector-valued outputs," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/b1eec33c726a60554bc78518d5f9b32c-Paper.pdf

[3] K. B. Peterson and M. S. Pederson, *The Matrix Cookbook*, 10 Nov. 2012, pp. 9–10.