ONLINE REGULARIZED DISCRIMINANT ANALYSIS

Umut Orhan, Ang Li and Deniz Erdogmus
Cognitive Systems Laboratory
Northeastern University
Boston, MA, USA

ABSTRACT

Learning the signal statistics and calibration are essential procedures for supervised machine learning algorithms. For some applications, e.g. ERP based brain computer interfaces, it might be important to reduce the duration of the calibration, especially for the ones requiring frequent training of the classifiers. However simply decreasing the number of calibration samples would decrease the performance of the algorithm if the algorithm suffers from curse of dimensionality or low signal to noise ratio. As a remedy, we propose estimating the performance of the algorithm during the calibration in an online manner, which would allow us to terminate the calibration session if required. Consequently, early termination means a reduction in time spent. In this paper, we present an updating algorithm for regularized discriminant analysis (RDA) to modify the classifier using the new supervised data collected. The proposed procedure consierably reduces the time required for updating the RDA classifiers compared to recalibrating them, that would make the performance estimation applicable in real time.

Index Terms— Pattern recognition, adaptive learning, brain computer interfaces, event related potential

1. INTRODUCTION

Most supervised machine learning applications consist of a learning or training data collection to calibrate the classifier or to learn the class distributions. Especially in some biomedical applications, signal statistics might change daily basis due to difference on application of the recording device or change in the signal. If the calibration is long in time, for reducing the calibration duration, it might become extremely important to estimate the sufficiency of the calibration data, give feedback and terminate the session early.

One of the important applications which brain computer interface (BCI) is a technology which is gradually arousing notice. In particular, when BCI-enabled devices are used as a communication system, it could be used by a considerable number of people with significant communication and motor disabilities to generate expressive language [1, 2, 3, 4]. Electroencephalography (EEG) based BCI is becoming particularly popular due to high temporal resolution and portability. Relevantly, event related potentials (ERP), especially P300, are used with paradigms with time locked stimuli. Most ERP and EEG based BCI systems require a calibration session before the actual use of the system, due to changes in signal statistics. The reasons might include, but not limited to, the difficulty of placing the EEG electrodes to the exact same location, change in skin resistance, daily attention level. Consequently, reducing the calibration duration while having enough samples to be able to learn the statistics becomes extremely important.

RSVP Keyboard™ is an EEG based BCI typing system [5], which utilizes rapid serial visual presentation (RSVP) to present temporally separated symbols. When the subject shows intent for a symbol in the randomly ordered sequence, the EEG responses corresponding to target and non-target responses becomes different. The intended symbol is decided via a binary classification using regularized discriminant analysis (RDA) on the extracted temporal features [6]. RDA and the updating algorithms proposed in this paper are not specific to RSVP Keyboard™, and may be applied to any ERP-based real time application.

RDA is a binary classifier, mostly used in situations where the number of samples is lower than the number of data dimensions. RDA requires estimation of means and covariances of the data vectors corresponding to two classes using a calibration session. To be able to give a performance estimate during calibration, the signal statistics are updated accordingly with the additional collected data samples with labels. In this paper, we propose an iterative method to update the performance estimate of an RDA classifier with lower computation time using the Woodbury matrix identity and the matrix determinant lemma.

2. REGULARIZED DISCRIMINANT ANALYSIS

RDA is a modified quadratic discriminant analysis (QDA) model. If the feature vectors corresponding to each class are assumed to have multivariate normal distribution, the Bayes optimal classifier minimizing the total risk resides in the QDA
model family [7]. Under this assumption, QDA depends on the inverse and the determinant of the class covariance matrices, which are generally estimated from a training data set. However the calculation of the inverse might be problematic due to singularities in the estimated covariance matrices for high dimensional small sample sized problems. As a solution to this problem, RDA applies regularization and shrinkage procedures to the estimated covariance matrices to eliminate the singularities [8]. The shrinkage operation makes the class covariances closer to an overall covariance, consequently to high dimensional small sample sized problems. As a solution the inverse and the determinant of the class covariance matrices, which are generally estimated from a training data set.

In most classification problems, the calibration of the classifier is accomplished using a fixed training dataset and training size is not adjustable during the calibration session. The problems which acquisition of new samples is slow and calibration sessions take long time would benefit greatly from an online performance tracking and early termination of the calibration session. This condition arises in most realtime ERP based BCI systems due to difficulty of collecting new ERP samples. In this paper, as a remedy to this problem, we propose low computational complexity updates for the updating the RDA classifier and K-fold cross validation based performance estimating methodology.

During K-fold cross validation, the calibration data collected until now is divided into K mutually exclusive sets, \( X = X_1 \cup X_2 \cup \cdots \cup X_K \), where \( n \) is total number of calibration samples. \( \forall k \in \{1, 2, \ldots, K\} \) and \( x \in X_k \), corresponding \( \hat{\mu}_c \) and \( \hat{\Sigma}_c(\lambda, \gamma) \) are estimated using the sample set \( X \setminus X_k \) and \( \delta_{RDA}(x) \) is calculated, consequently. The calculation of \( \hat{\Sigma}_c(\lambda, \gamma)^{-1} \) and \( \ln \det(\hat{\Sigma}_c(\lambda, \gamma)) \) is needed to calculate \( \delta_{RDA}(x) \). Since both matrix inversion and determinant calculation are computationally complex for high dimensions, it is practically not feasible to frequently retrain the classifiers after the addition of new samples. Therefore we propose updating the inverse and the determinant using lower complexity updates.

After a low sample initial calibration phase, which contains cross validation, we obtain K classifiers, containing the information of \( \hat{\Sigma}_c(\lambda, \gamma)^{-1} \) and \( \ln \det(\hat{\Sigma}_c(\lambda, \gamma)) \), where \( det(\cdot) \) represents the matrix determinant. When there exist new samples \( X'_n \) to update the classifier, we map each new sample to one of K cross validation partitions as an addition. \( \forall k \) and new sample set \( X'_k \) belonging to partition \( k \), we update the classifiers indexed as \( \{1, 2, \ldots, K\} \setminus \{k\} \) using \( X'_k \). The update of means and priors is trivial. Update of \( \hat{\Sigma}_c(\lambda, \gamma)^{-1} \) and \( \ln \det(\hat{\Sigma}_c(\lambda, \gamma)) \) for one fold, or one classifier, is explained in 3.1 and 3.2. After the updates \( \delta_{RDA}(x) \) for all samples including the old samples is recalculated, since the change of the classifier also changes \( \delta_{RDA}(x) \). The calculation of new scores immediately gives us an estimate of performance via the calculation of the area under ROC curve.

3. ONLINE RDA

In most classification problems, the calibration of the classifier is accomplished using a fixed training dataset and training

\[
\hat{\Sigma}_c(\lambda) = \frac{\hat{S}_c(\lambda)}{W_c(\lambda)},
\]

\[
\hat{S}_c(\lambda) = (1 - \lambda) \hat{S}_c + \lambda \hat{S},
\]

\[
W_c(\lambda) = (1 - \lambda) N_c + \lambda N,
\]

\[
\hat{S}_c = N_c \hat{\Sigma}_c,
\]

where \( \hat{\Sigma}_c \) is the sample covariance estimated using \( N_c \) samples for class \( c \in \{0, 1\} \) and \( \lambda \) is the shrinkage parameter. Shrinkage procedure makes the covariance matrices closer to each other, e.g \( \lambda = 1 \) corresponds to equal covariance case. Additionally a regularization procedure is applied as

\[
\hat{\Sigma}_c(\lambda, \gamma) = (1 - \gamma) \hat{\Sigma}_c(\lambda) + \frac{1}{p} tr[\hat{\Sigma}_c(\lambda)] I
\]

where \( \gamma \) is the regularization parameter, \( p \) is the number of dimensions, \( tr[\cdot] \) is the trace function and \( I \) is identity matrix. This procedure makes the class covariances closer to circular, e.g \( \gamma = 1 \) case corresponds to circular covariances.

Using the modified estimates of covariance matrices obtained after shrinkage and regularization operation, the optimal Bayesian decision rule becomes

\[
\delta_{RDA}(x) = \ln \frac{f_N(x; \hat{\mu}_c, \hat{\Sigma}_1(\lambda, \gamma)) \hat{\pi}_1}{f_N(x; \hat{\mu}_0, \hat{\Sigma}_0(\lambda, \gamma)) \hat{\pi}_0} \geq \tau,
\]

where \( \hat{\mu}_c, \hat{\pi}_c \) are estimates of class means and priors respectively; \( x \) is the data vector to be classified. \( f_N(x; \mu, \Sigma) \) is the pdf of a multivariate normal distribution and \( \tau \) is a threshold to be decided based on risks. Using a test data set or applying cross validation, a receiver operating characteristics (ROC) curve representing the false positive and true positive rates for various selections of the thresholds might be drawn. Furthermore, the area under the ROC curve may be used as a performance estimate of the classifier. This performance measure can be loosely used to decide if the proposed RDA model fits the data.

3. ONLINE RDA

In most classification problems, the calibration of the classifier is accomplished using a fixed training dataset and training
where $\mu_n$ is the mean estimated using $\{x_1, x_2, \ldots, x_n\}$. Consequently $S_{c,n+m}$ becomes

$$
\hat{S}_{c,n+m} = \sum_{i=1}^{n+m} (x_i - \hat{\mu}_{n+m}) (x_i - \hat{\mu}_{n+m})^T
$$

$$
= \sum_{i=1}^{n+m} x_i x_i^T - (n + m) \hat{\mu}_{n+m} \hat{\mu}_{n+m}^T
$$

$$
= \hat{S}_{c,n} + n \hat{\mu}_{n} \hat{\mu}_{n}^T + \sum_{i=n+1}^{n+m} x_i x_i^T - (n + m) \hat{\mu}_{n+m} \hat{\mu}_{n+m}^T
$$

$$
= \hat{S}_{c,n} + Y_{c,n} Z_{c,n}^T
$$

where

$$
Y_{c,n} = [x_{c,n+1}, \ldots, x_{c,n+m}, n \hat{\mu}_n, - (n + m) \hat{\mu}_{n+m}]$$

$$
Z_{c,n} = [x_{c,n+1}, \ldots, x_{c,n+m}, \hat{\mu}_n, \hat{\mu}_{n+m}]
$$

and $\hat{\mu}_{n+m} = \frac{1}{n+m} \left( \hat{\mu}_n + \sum_{i=n+1}^{n+m} x_i \right)$. Applying the shrinkage procedure using (1) and (2),

$$
S_{c,n} (\lambda) = (1 - \lambda) S_{c,n} + \lambda S_n
$$

where $S_n = S_{0,n} + S_{1,n}$. Consequently, $\hat{S}_{c,n+m}$ becomes

$$
S_{c,n+m} (\lambda) = (1 - \lambda) S_{c,n+m} + \lambda S_{n+m}
$$

$$
= S_{c,n} (\lambda) + U_c V_c^T
$$

where

$$
U_0 = [Y_0, \lambda Y_1], V_0 = [Z_0, Z_1]
$$

$$
U_1 = [Y_1, \lambda Y_0], V_1 = [Z_1, Z_0].
$$

Thereafter, the inverse of the shrunk covariance matrix estimates can be rewritten by (3) as,

$$
\Sigma_{c,n+m}^{-1} (\lambda) = W_{c,n+m}^{-1} (\lambda) S_{c,n+m}^{-1} (\lambda).
$$

At this point we can use Woodbury Matrix Identity [9] on (5), stated as following,

$$
(A + UCV^T)^{-1} = A^{-1} - A^{-1} U \left( C^{-1} + V^T A^{-1} U \right)^{-1} V^T A^{-1}
$$

where $A$ and $C$ are $p$-by-$p$ and $k$-by-$k$ invertible matrices, respectively, and $U$ and $V$ are $p$-by-$k$ and $k$-by-$p$ matrices, respectively; to obtain

$$
\Sigma_{c,n+m}^{-1} (\lambda) = S_{c,n}^{-1} (\lambda) + S_{c,n}^{-1} (\lambda) U_c
$$

$$
\cdot \left( I_p + V_c S_{c,n}^{-1} (\lambda) U_c \right)^{-1} V_c S_{c,n}^{-1} (\lambda).
$$

This recursive formula allows us to update the inverse of the covariance matrix after shrinkage operation using the new samples with a lower computation time.

### 3.2. Updating the determinant

Similar to 3.1, we can update the determinant of the shrunk covariance matrix via Matrix Determinant Lemma [10],

$$
\det (A + UV^T) = \det (I + V^T A^{-1} U) \det (A).
$$

Consequently, from (5), we obtain

$$
\ln (\det (S_{c,n+m+1}^{-1} (\lambda))) = - \ln (\det (I_p + V_c S_{c,n}^{-1} (\lambda) U_c))
$$

$$
+ \ln (\det (S_{c,n}^{-1} (\lambda))).
$$

From (6), we obtain the final shrinkage update formula

$$
\ln (\det (\Sigma_{c,n+m+1}^{-1} (\lambda))) = \ln (\det (S_{c,n+m}^{-1} (\lambda)))
$$

$$
+ p \ln (W_{c,n+m} (\lambda)).
$$

The regularization procedure is only administered at the initialization part and it hasn’t been included into updating processes.

The new scores for all samples, i.e $\delta_{RDA}(x)$ are calculated using the recursive inverse covariance and determinant updates of the shrinkage matrices obtained. Then these scores can be used to calculate the performance estimate based on area under ROC curve.

### 4. EXPERIMENTAL RESULTS

For the experimental results, the calibration sessions collected from two healthy subjects and one subject with locked-in-syndrome (LIS) using the RSVP Keyboard™ system are used to analyze the performance of the algorithm in an offline analysis. In the calibration session of RSVP Keyboard™, a target symbol is randomly selected among 26 letters in the English alphabet, a backspace symbol and a space symbol. At each epoch the selected target symbol, which the subject expected to pay attention to, and a fixation screen are shown followed by 10 randomly ordered symbols. The series of symbols are selected randomly with replacement while ensuring the existence of the target symbol, and presented sequentially with 400 ms inter stimulus interval. At each calibration session, 50 or 75 epochs, which is decided prior to session, are shown.

The signals are recorded using a g.USBamp biosignal amplifier using active g.Butterfly electrodes from G.tec (Graz, Austria) at 256Hz. The EEG channels positioned according to the International 10/20 System were Fp1, Fp2, F3, F4, Fz, Fc1, Fc2, Cz, P1, P2, C1, C2, Cp3, Cp4, P5, P6. Signals were filtered by nonlinear-phase 0.5-60 Hz bandpass filter and 60 Hz notch filter (G.tec’s built-in design), afterwards signals filtered further by 1.5-42 Hz linear-phase bandpass filter. The filtered signals were downsampled to 128Hz. For each channel, stimulus-onset-locked time windows of [0,500]ms following each symbol onset was taken as the stimulus response.

For the analysis of the algorithm, the session is divided into two parts; initialization and updating. In the initialization section, the initial classifiers are trained with three different methods to select classifier parameters $\lambda$ and $\gamma$. 
Fig. 1. Progress of AUC for Healthy Subject 1

Fig. 2. Progress of AUC for Healthy Subject 2

Fig. 3. Progress of AUC for Subject with LIS
• Predefined set of parameters with $\{\lambda, \gamma\} = \{0.9, 0.1\}$: This heuristic selection of parameters makes class covariances close to each other with a small regularization and allows a quick selection.

• Grid search

• Nelder-Mead simplex-reflection method

Since the initialization set is small, gamma selected from the initialization set might be larger than the gamma which would be selected afterwards. Therefore during initialization, we only search gamma up to 0.2. In the updating section of the session, each epoch is considered as a new data to update the classifiers and classifiers are updated sequentially.

As a practical example, the number of epochs in the initialization section is selected as 25, and the number of folds in K-fold cross validation is selected as 10. The analysis procedure is explicitly following, using the first 25 epochs a cross validation is applied to train 10 classifiers and classifier parameters, $(\lambda, \gamma)$ are selected. For each new epoch three possible approaches are investigated,

• Without updating: The classifiers are not modified, and the scores for the new epoch are calculated. Since there is no update on classifier, doesn’t take much time.

• Retraining: The classifiers are trained again using all existing data, and the scores for all of the epochs are calculated. Since it retrains the classifier it is computationally complex. Probably not applicable in real time.

• Updating: The classifiers are updated using the proposed online RDA algorithm, and the scores for all epochs are calculated.

From Fig. 1 to Fig. 3, the progress of AUCs is shown under different initializations of classifier parameters and for different update approaches for each new epoch. In all cases the end AUC of updating via online RDA and retraining are close to each other, but retraining is greater, without updating approach was always worse than the others. Fig.4 represents the time spent in the updating process. Fig.4 indicates that online RDA costs less time than retraining with a similar AUC estimate.

5. CONCLUSION

In this paper, we presented a technique that uses Woodbury matrix identity and matrix determinant lemma to update the RDA model. This update allows one to update classifiers in K-fold cross validation and consequently estimate of the performance during the calibration session for real time systems that uses RDA classifier. This approach will be incorporated to RSVP Keyboard™ in future for the purposes of giving feedback and terminating the calibration session earlier. The proposed algorithm costs less time than retraining the classifiers, which would allow the performance estimate to be feasible in real time. Moreover, online RDA can be used for active learning, unsupervised or using data driven labels, in the testing mode or spelling mode of RSVP Keyboard™. This might even allow algorithm to accommodate tiredness or the change in signal statistics.

In the algorithm proposed, the regularization operation is ignored during the update, which causes a performance difference between retraining of the classifiers and updating of the classifiers during the update portion. If a method to update the regularization operation is found, this would lead to an exact match with the retraining result of the classifier and potentially result in a more accurate estimate of the performance.

As we have more samples and the assumption on the class distribution family is correct, the signal statistics, i.e. means and covariances, will get asymptotically closer to the real statistics. Therefore one might expect the performance to increase with every update iteration. However it might not be the case in the performance of the RDA classifier in BCIs. Firstly, due to shrinkage and regularization operations, the regularized covariance matrices might not asymptotically approach to the true covariance matrices. Secondly, the samples might not be i.i.d. due to various reasons, i.e. attention of the subject.

6. ACKNOWLEDGMENTS

This research is supported by grants NIH R01 DC009834, NSF IIS-0914808, NSF CNS-1136027, and NSF IIS-1149570.

7. REFERENCES


