34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Sparse Linear Regression with Elastic Net Regularization for Brain-Computer Interfaces

John W. Kelly, *Student Member, IEEE*, Alan D. Degenhart,
Daniel P. Siewiorek, *Fellow, IEEE*, Asim Smailagic, *Fellow, IEEE*, Wei Wang

***Abstract*— This paper demonstrates the feasibility of decoding neuronal population signals using a sparse linear regression model with an elastic net penalty. In offline analysis of real electrocorticographic (ECoG) neural data the elastic net achieved a timepoint decoding accuracy of 95% for classifying hand grasps vs. rest, and 82% for moving a cursor in 1-D space towards a target. These results were superior to those obtained using ℓ_2-penalized and unpenalized linear regression, and marginally better than ℓ_1-penalized regression. Elastic net and the ℓ_1-penalty also produced sparse feature sets, but the elastic net did not eliminate correlated features, which could result in a more stable decoder for brain-computer interfaces.**

Index Terms - elastic net, sparse linear regression, feature selection, neural signals, brain-computer interfaces

I. INTRODUCTION

Brain-computer interfaces (BCIs) have progressed greatly in recent years, and continue to move towards the goal of offering neural control of assistive devices. This progress is due in part to improvements in computing power, as well as recording and processing methods, that allow increasingly larger amounts of neural data to be processed and analyzed in real time. This increase in data generally increases the likelihood that useful signals will be present, but it also causes an increase in the amount of irrelevant or noisy data.

In the case of a BCI the goal is typically to decode the neural data to produce a control signal for an external device such as a computer cursor, robotic arm, or wheelchair [1]. If not handled properly, extraneous or contaminated neural features can translate into noise in the output. The objective then should be to implement a neural decoding method that is invariant to irrelevant features.

One strategy is to observe the modulation of the neural signals and then choose appropriate parameters for the decoder [2]. Neural plasticity then typically allows the brain to further adapt to the selection [3]. This strategy has even been taken to the extreme in non-human primates, where it was shown that the brain could eventually adapt to randomly chosen features [4]. This strategy is time-consuming, though, and requires operation by highly trained personnel.

This material is based upon work supported by a National Science Foundation Graduate Research Fellowship under Grant No. 0750271, and the Quality of Life Technology Center under NSF Grant No. EEEC-0540865

J. W. Kelly, D. P. Siewiorek, and A. Smailagic are with the Dept. of Electrical and Comp. Eng., Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail:jwkelly@ece.cmu.edu; dps@cs.cmu.edu; asim@cs.cmu.edu)

A. D. Degenhart is with the Dept. of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15213 USA (e-mail: add19@pitt.edu)

W. Wang is with the Dept. of Physical Medicine and Rehab., University of Pittsburgh, Pittsburgh, PA 15213 USA (e-mail: wangwei3@pitt.edu)

For BCIs to be usable by non-experts, the decoding algorithm needs to be highly automated and robust. This paper investigates the use of elastic net regularization for linear regression in BCIs. The next section covers the background of this technique, as well as a few related methods. The elastic net is also compared to some of these other methods through the decoding of offline electrocorticography (ECoG) neural data and the results are discussed with their possible implications towards online BCI decoding.

II. BACKGROUND

A. The Curse of Dimensionality

The task of training a classifier with a large number of irrelevant features and a small number of observations is not unique to BCIs. Overfitting and the contributions of noisy features both become major concerns in these types of problems. Dimensionality reduction, feature selection, and regularization are methods often employed in high-dimensional decoding problems. Dimensionality reduction methods such as principal component analysis (PCA) and linear discriminant analysis (LDA) have been used in BCIs [5], but these techniques transform the features to a new basis, making it more difficult to interpret the real-world significance of the raw features. Feature selection retains the original basis, but might not capture as much of the original information as dimensionality reduction.

A number of BCI studies have used 'pure' feature selection methods such as forward stepwise regression that only choose features and then solve for weights using a standard method such as ordinary least squares (OLS). Forward stepwise regression adds the feature at each step that eliminates the most residual error (backward stepwise removes features at each step that eliminate the least amount of error). These techniques can be biased, since the best set of $M+1$ features does not necessarily contain the best set of M features [6]. They can also be unstable, in that a small change in the features could result in a large change in the output. Stagewise regression attempts to minimize the problems associated with stepwise regression by increasing a feature's weight by a small amount at each step rather than all the way to the least squares solution. The adjusted feature could remain the same for multiple steps.

B. Regularized Linear Regression

Linear regression, which has been used frequently in BCIs, takes the form of the optimiztion problem given by (1). Y is a vector containing N observations, $\mathbf{X}$ is an $N \times M$ matrix

containing M features for each observation, β is a vector of M weights that map the features to the observations, and θ is the bias, or offset, term. Although there are many methods for computing β based on this model, the simplest is OLS.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \|Y - \mathbf{X}\beta + \theta\|_2^2 \quad (1)$$

Regularization is a common way of addressing overfitting and other problems with high-dimensional feature spaces. This technique adds a penalty term, represented by $\mathrm{c}(\beta)$ in (2). λ is a free parameter that determines the magnitude of the penalty. In the case of ℓ_2 regularization the penalty is the ℓ_2-norm of β, and in ℓ_1-regularization the penalty is the ℓ_1-norm. The use of these penalties is sometimes referred to as ridge regression and lasso (least absolute shrinkage and selection operator), respectively. The lasso penalty is more computationally challenging since it is non-differentiable, but it also performs feature selection by reducing some values of β to zero [7]. It has actually been shown that in stagewise regression as the feature weight stepsize optimally approaches zero, the result approaches the lasso [6].

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} (\|Y - \mathbf{X}\beta\|_2^2 + \lambda * \mathrm{c}(\beta)) \quad (2)$$

C. Elastic Net

The lasso has proven to be be highly effective in classification problems with a large number of irrelevant features and has been used on neural data, however, it is not without drawbacks. In a situation where multiple features are useful but highly correlated, lasso tends to keep one and drop the rest. Stability then becomes a concern, and robustness could also be an issue in situations where not all features remain reliable over time due to noise or other events.

Elastic net regression attempts to address these concerns by blending the ℓ_1 and ℓ_2 penalties, as shown in (3). The goal in elastic net is to produce a sparse feature space with the ℓ_1 penalty, but improve stability and retain correlated features with the ℓ_2 penalty. Like the lasso this is not a computationally simple problem, but efficient methods for solving it have been developed. There is also an additional free parameter in α, which determines the relative strength of the penalties. Previous studies have shown this technique to be effective in classification of functional magnetic resonance imaging (fMRI) data [8], [9].

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} (\|Y - \mathbf{X}\beta\|_2^2 + \lambda(\alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2)) \quad (3)$$

III. Methods

A. Data

The datasets used in this study consisted of electrocorticographic (ECoG) signals recorded from two subjects undergoing monitoring for intractable epilepsy. Informed consent was obtained from both subjects prior to testing, and all data collection and experimental procedures were approved by the Institutional Review Board of the University of Pittsburgh.

The signals were sampled at 1200 Hz and bandpass filtered from 0.1 to 200 Hz using g.USBamp amplifiers. The data signals were acquired using BCI2000, an open-source software package written in C++ [10], and were then sent to Craniux, an open-source software package created in LabVIEW that handled experimental parameters and execution [11]. Spectral estimation was performed using the maximum entropy method with 10 Hz frequency bins, 300 ms windows, and a 33 ms step size. Subjects were observed to ensure eye and facial movements were not used to control the BCI.

In choosing the data for offline analysis, it was attempted to use experimental paradigms that have minimal online error correction by the user. Paradigms with error correction, such as a 2-dimensional cursor task in which the subject might not move along the ideal path to the target, present problems in offline analysis. It can become difficult to determine the subject's exact intent and to incorporate the neural adaptation that is occurring as a result of error correction.

The experimental paradigm with Subject A was a simple hand grasp screening task. The subject was presented with a visual cue in the form of a gray box on a black screen, and was instructed to continually open and close the hand while the cue was present. The hand performing the grasps was contralateral to grid placement.

For Subject B, the experimental paradigm was a 1-dimensional center-out cursor task. A cursor would appear on the screen along with a target to the right or the left of the cursor. The subject was instructed to perform hand grasps to move the cursor to the right, and to move their elbow to send the cursor to the left. The cursor was constrained to horizontal movement and the trial ended when the cursor touched the target. In this way it is assured that the subject was always attempting to move the cursor in the same direction for the duration of each trial. The hand and elbow used for movement were again contralateral to grid placement.

Subject A had 64 recorded channels: 48 from a standard clinical ECoG grid, and 16 from a high-density ECoG research grid. Subject B had 128 recorded channels: 62 from clinical grids, 32 from research grids, 2 EKG channels, and 32 open channels. The EKG and open channels were left in the data because part of the goal was to show the feasibility of an automated decoder with no supervision on channel selection. Data from Subject A consisted of 5 sessions with 24 trials each. For Subject B, data contained 4 sessions with 42 to 90 trials each for a total of 234 trials.

B. Classification

Decoding of the neural signals was done in an offline analysis using four different methods: elastic net, lasso, ridge regression, and OLS. The solutions for the first three methods were calculated using a modified version of glmnet, a freely available software package developed at Stanford University. Glmnet uses cyclical coordinate descent in a pathwise fashion and has previously shown excellent results and convergence speed [12], [13].

To determine the best value of λ for elastic net, lasso, and ridge regression, 10-fold cross-validation was performed for

each training of the classifier across 20 different values of λ. A similar scheme was originally adopted to determine the best value of α, but it was found that this method generally caused the result to closely mirror the lasso solution. While this solution may indeed be the best fit for a particular set of training data, it fails to produce the robustness and stability that were earlier discussed as motivations for using the elastic net penalty. For this reason, α was set at 0.1 ($\alpha = 1$ is equivalent to lasso and $\alpha = 0$ is ridge regression).

Decoding was done on each session using 10-fold cross-validation. Mean and standard deviation were calculated from the training set to normalize all features. In the training set, the time-average of each feature over each trial was used, but in the testing set the decoding was done on each timepoint of spectral data as it would be in a real-time BCI. Results were averaged across all timepoints for each subject.

The main metric calculated was the percent of timepoints in which the decoder was correct. For Subject A this means determining whether the subject was grasping or not, and for Subject B this means determining if the cursor would move in the correct direction. Since this metric only determines the accuracy of the direction of movement and not magnitude, the change in distance to target was also measured for Subject B. For both subjects, timepoints that were within 500 ms of stimulus onset were ignored. This was to ensure that the spectral estimation window consisted of neural data produced after the subject had reacted to the stimulus.

IV. Results and Discussion

Fig. 1 shows the percentage of timepoints that were classified incorrectly for both subjects using each decoder. Elastic net had a lower error than the other decoders across all sessions for both subjects. The advantage over lasso in the average error is small, although elastic net did appear to be more dependable across sessions as indicated by the maximum session error for both subjects. The range of error across sessions was quite large for all decoders with Subject B, but as expected the error and consistency for the simpler task performed by Subject A was much better.

The better consistency of Subject A helped highlight the significant improvement of elastic net and lasso over ridge regression and OLS ($p < 0.05$ for all cases). It should also be noted that for similar tasks results are often reported for testing on time-averaged features for each trial rather than individual timepoints, which generally results in lower errors. For Subject B this method resulted in errors of 10%, 13%, 22%, and 20% (elastic net, lasso, ridge regression, OLS).

The results in Fig. 2 reinforce those given by Fig. 1. Additionally, it shows that ridge regression produced a control signal that, although not as accurate on average, was much more stable than the other decoders in that it never moved the cursor a great distance in either direction. This could be desirable in operating physical devices such as robotic arms where sudden jerks and unpredictability could present a danger. The elastic net result demonstrates this same property to some extent with a distribution that has slightly lower variance than lasso and OLS, but with

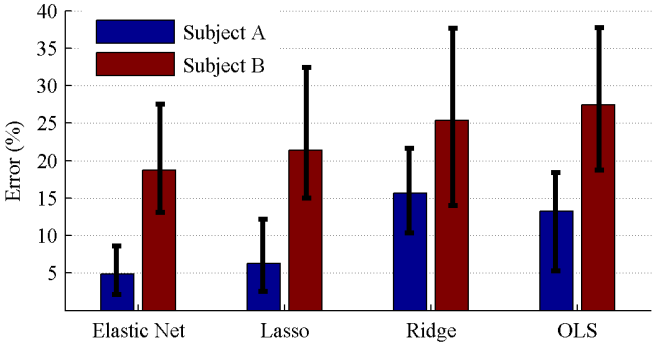

Fig. 1. Percentage of timepoints classified incorrectly. The value shown indicates the percentage across all timepoints in all sessions. The error bars indicate the minimum and maximum percentage of timepoints classified incorrectly for an individual session.

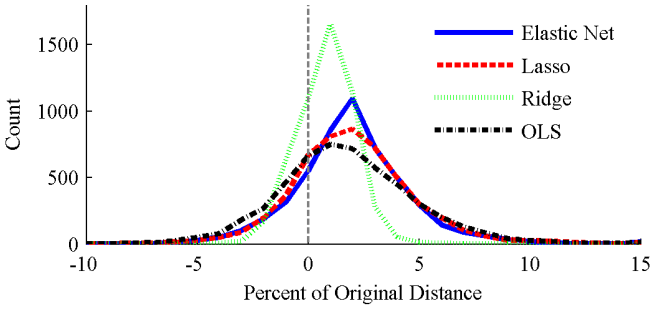

Fig. 2. Change in distance to target for 1D cursor control. The distances were normalized by the original distance to the target, and then binned with a bin width of 1. The values represent the count of timepoints in each bin.

a higher mean than ridge regression. A higher variance in this distribution could be an indication of more noise being factored into the decoding results.

The sparsity of the weights used by the decoders is also important in their discussion. Fig. 3 shows the weights calculated by each decoder when trained on one session of data from Subject B. As expected OLS has no sparsity in its results and ridge regression, while having many weights that are close to zero, also does not give a sparse set of weights. Some banding can even be seen in these weights near 120 Hz and 180 Hz, which is most likely the result of line noise harmonics in the data. Lasso, on the other hand, produces a set of weights in which only 35 of the 2,560 weights are non-zero. For the elastic net decoder, 114 features had non-zero weights. Elastic net and lasso also chose no features from the 32 open channels at the end, although there were a few small non-zero weights on one of the EKG channels. If these decoders were truly used in an automated system, steps would need to be taken to address the presence of artifacts in the signals (such as eye movement) that could be modulated by the user to control the BCI.

Many of the features eliminated by lasso but retained by elastic net closely neighbor a non-zero lasso weight both spatially and in frequency. For example, lasso used the 100-110 Hz bin on channel 27. Elastic net used this feature as well as the rest of the gamma band on channel 27 and channel 19, which was spatially adjacent. This extra

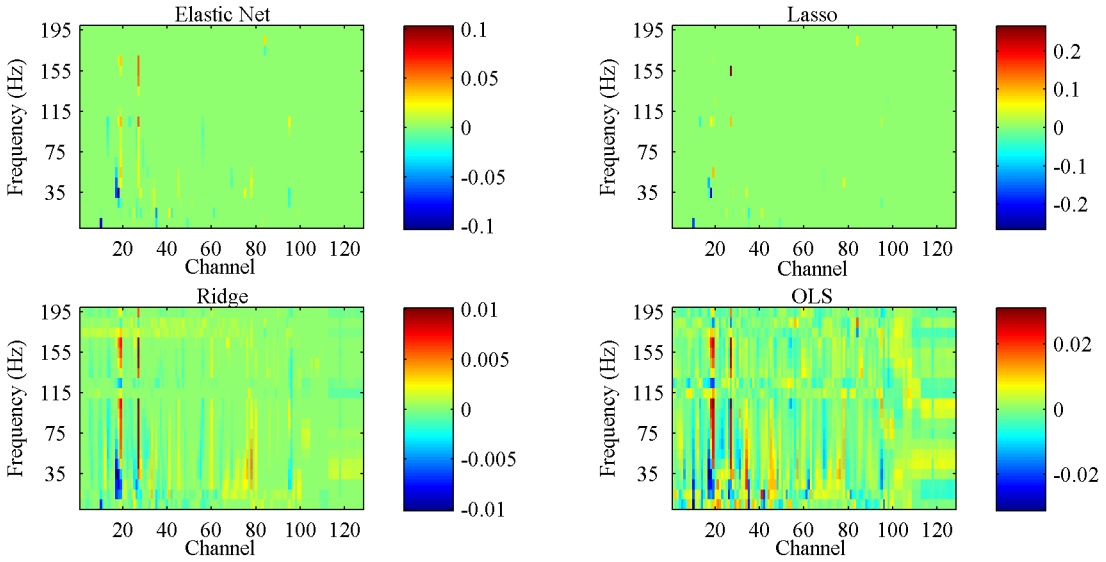

Fig. 3. Decoding weights calculated from one session of data from Subject B. Note that the intensity scale is not the same for each decoder to help ensure that the non-zero weights are properly highlighted in the sparse solutions.

reduncancy in the decoder could be useful in a BCI where the features are subject to noise and are themselves adapting due to neural plasticity. The extra stability in the feature set would also be desired when re-training so that the decoding weights aren't as much of a moving target for the BCI user. When trained on each session of Subject B data, not a single feature was common across all sessions for lasso.

V. CONCLUSIONS

The results here have demonstrated the feasibility of sparse linear regression using the elastic net penalty for BCIs. The decoding accuracy of this method was significantly better than ridge regression and OLS, but only slightly better than lasso. The feature set the elastic net chose appeared to retain more correlated features than the lasso, though, resulting in a more stable set of feature weights across training sessions.

Some level of sparsity should be desired in nearly any decoding problem with a high-dimensional feature set in order to eliminate noisy and irrelevant features, but the proper level of sparsity in a BCI remains an open problem. Having fewer features may allow the BCI user to more easily adapt to the decoding weights. Eliminating features that are only moderately useful could allow those features to be used to control an additional degree of freedom. As discussed here, though, a feature space that is too sparse could result in loss of robustness and stability. A further advantage of the elastic net penalty is that the level of sparsity can be scaled all the way from the lasso to the ridge regression solution.

This study has provided initial results to show the usefulness of the elastic net penalty in BCIs. To fully measure its effectiveness, though, as well as to determine the proper feature set sparsity for BCI decoding, further studies should be done with online decoding of neural signals. This allows the BCI user's adaptation to become part of the equation.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control." *Clin. Neurophys.*, vol. 113, no. 6, pp. 767–91, Jun. 2002.

[2] W. Wang *et al.*, "Human Motor Cortical Activity Recorded with Micro-ECoG Electrodes, During Individual Finger Movements," in *Conf. of the IEEE Eng. in Med. and Bio. Soc.*, vol. 2009, Jan. 2009, pp. 586–9.

[3] W. Wang *et al.*, "Neural interface technology for rehabilitation: exploiting and promoting neuroplasticity," *Phys. Med. and Rehab. Clinics of North Amer.*, vol. 21, no. 1, pp. 157–178, 2010.

[4] A. G. Rouse and D. W. Moran, "Neural adaptation of epidural electrocorticographic (EECoG) signals during closed-loop brain computer interface (BCI) tasks." in *Conf. of the IEEE Eng. in Med. and Bio. Soc.*, vol. 2009, Jan. 2009, pp. 5514–7.

[5] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals." *J. of Neural Eng.*, vol. 4, no. 2, pp. R32–57, Jun. 2007.

[6] T. Hesterberg, N. H. Choi, L. Meier, and C. Fraley, "Least angle and l1 penalized regression: A review," *Statistics Surveys*, vol. 2, pp. 61–93, 2008.

[7] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[8] M. K. Carroll, G. a. Cecchi, I. Rish, R. Garg, and a. R. Rao, "Prediction and interpretation of distributed neural activity with sparse models," *NeuroImage*, vol. 44, no. 1, pp. 112–22, Jan. 2009.

[9] S. Ryali, K. Supekar, D. a. Abrams, and V. Menon, "Sparse logistic regression for whole-brain classification of fMRI data." *NeuroImage*, vol. 51, no. 2, pp. 752–64, Jun. 2010.

[10] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system." *IEEE T. on Biomed. Eng.*, vol. 51, no. 6, pp. 1034–43, Jun. 2004.

[11] A. D. Degenhart *et al.*, "Craniux: A LabVIEW-Based Modular Software Framework for Brain-Machine Interface Research." *Comp. Intell. and Neuroscience*, vol. 2011, Jan. 2011.

[12] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent." *J. of Stat. Software*, vol. 33, no. 1, pp. 1–22, Jan. 2010.

[13] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for cox's proportional hazards model via coordinate descent," *J. of Stat. Software*, vol. 39, no. 5, 2011.