Support Vector Machines (SVMs) and EEG Analysis

1 Support Vector Machines (SVMs)

Classifier learning can be seen as finding a curve that best separates two sets of points. The goal is to find an **optimal curve** out of infinitely many possibilities. However, defining "optimal" is challenging. For example, if we define the optimal curve as the one that minimizes mistakes within the two sets, it may not generalize well to unseen data—a problem known as **overfitting**.

Support Vector Machines (SVMs) address overfitting by focusing on the points in the dataset that are closest to the line separating the two classes; these points are called **support vectors**. SVMs maximize the **margin** (distance) between the support vectors and the separating line. After finding this line using the margins, the SVM creates a formula called the **weight vector**, which indicates the importance of each feature in deciding the classification.

For classification, the features of an object are plugged into the weight vector to get a score:

- A **positive score** indicates one class.
- A **negative score** indicates the other class.

1.1 Feature Scaling

Before training, it's essential to ensure that different features are on a similar scale to prevent bias from features with large numbers. This is typically done by:

1. Subtracting the mean of each feature.
2. Dividing by the standard deviation of each feature.

2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique used for dimensionality reduction and feature extraction. It transforms data into a lower-dimensional

space while preserving the most important patterns or "components." In EEG classification, PCA helps reduce the complexity of high-dimensional data while retaining the most informative features for classification tasks like P300 detection.

2.1 Steps in PCA

PCA involves the following steps:

1. **Standardization**: Center the data by subtracting the mean and scaling to unit variance for each feature.
2. **Covariance Matrix Computation**: Calculate the covariance matrix to understand how different features vary with respect to each other.
3. **Eigenvalue and Eigenvector Calculation**: Compute the eigenvalues and eigenvectors of the covariance matrix. The eigenvectors represent directions (principal components), and the eigenvalues represent the magnitude of variance along those directions.
4. **Select Principal Components**: Sort eigenvectors by their eigenvalues in descending order and select the top components that capture the majority of the variance.
5. **Transform the Data**: Project the original data onto the selected principal components, resulting in a reduced feature set.

2.2 Benefits of PCA in EEG Analysis

Using PCA in EEG analysis, specifically for P300 classification, provides several benefits:

- **Noise Reduction**: By focusing on the components with the most variance, PCA can help eliminate noise and irrelevant features from EEG data.
- **Dimensionality Reduction**: EEG data often has high dimensionality (many channels and time points), which can be computationally expensive. PCA reduces the dimensionality, making it more manageable for machine learning algorithms like SVM.
- **Improved Classifier Performance**: Reducing irrelevant features can improve the accuracy and generalization of classifiers by focusing on the most informative patterns.

3 Application of SVMs in EEG-Based P300 Detection

In EEG-based Brain-Computer Interfaces (BCIs), detecting the P300 event-related potential is crucial for identifying responses to stimuli. This section summarizes key steps in applying SVMs to classify P300 responses in EEG data.

3.1 Data Preprocessing

To extract meaningful features from EEG data, the following steps are typically applied:

1. **Bandpass Filtering**: Apply a bandpass filter to remove noise outside the target frequency range.
2. **Epoch Extraction**: Extract time segments (epochs) aligned with stimuli to capture P300 responses.
3. **Feature Extraction with PCA**: Use Principal Component Analysis (PCA) to reduce the dimensionality of the data while retaining the most informative components.

3.2 Feature Selection

After feature extraction, a feature selection step is applied to obtain a subset of features that may improve classifier performance. Reducing the number of features reduces the number of parameters the classifier needs to optimize, which can lead to better efficiency and effectiveness. The main feature selection approaches are as follows:

1. **Filter Methods**: These methods rely on measures of the relationship between each feature and the target class, independently of the classifier to be used. The coefficient of determination (the square of the Pearson correlation coefficient) can serve as a feature ranking criterion. However, the correlation coefficient detects only linear dependencies between features and classes. To exploit non-linear relationships, one solution is applying non-linear preprocessing, such as squaring or taking the logarithm of features.
2. **Wrapper Methods**: In this approach, a subset of features is selected and presented as input to a classifier for training. The resulting performance is observed, and the search stops when a specified criterion is satisfied. Otherwise, a new subset is proposed.
3. **Embedded Methods**: These methods integrate feature selection and evaluation into a single process, such as with a decision tree or a multilayer perceptron with optimal pruning strategies.

Five feature selection methods were evaluated on the BCI Competition III datasets, namely information gain ranking, correlation-based feature selection, Relief, consistency-based feature selection, and 1R ranking. Among ten classifiers, the top three feature selection methods were correlation-based feature selection, information gain, and 1R ranking.

3.3 Correlation-based Feature Selection

Correlation-based Feature Selection (CFS) is a technique that identifies subsets of features that have high correlation with the target variable but low correlation with each other. The purpose of CFS is to select a subset of features that provides the maximum information about the target while minimizing redundancy among the features.

A CFS algorithm calculates the correlation between each feature and the target variable. It then computes the correlation between each pair of features and selects the subset with the highest correlation with the target variable and the lowest correlation with each other.

In the following code, CFS is applied to a Breast Cancer diagnostic dataset containing 569 samples of malignant and benign tumor cells. The goal is to determine if a tumor is malignant or benign based on features such as radius, texture, and perimeter.

The first part of the code imports the necessary libraries and loads the dataset:

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
```

Next, it calculates the correlation matrix between the features and the target variable:

```
corr_matrix = df.corr()
corr_with_target = corr_matrix['target']
```

The code then sorts the values in descending order, selects the top k features, and checks the correlation matrix among the selected features:

```
selected_corr_matrix = selected_features.corr()
selected_corr_matrix
```

3.4 Training and Classification with SVM

After preprocessing, the reduced feature set is used to train an SVM classifier to differentiate between target and non-target responses. The classifier's goal

is to maximize the separation (margin) between P300 and non-P300 responses, enhancing single-trial classification accuracy.

4 References

- Bernard Ng, Rebecca K. Reh, Sara Mostafavi, *A Practical Guide to Applying Machine Learning to Infant EEG Data*, Developmental Cognitive Neuroscience, Volume 54, 2022, 101096, ISSN 1878-9293. https://doi.org/10.1016/j.dcn.2022.101096
- Nand Sharma, *Single-trial P300 Classification using PCA with LDA, QDA, and Neural Networks*, arXiv preprint, 2017. https://arxiv.org/pdf/1712.01977
- Grosse-Wentrup, M., *The Role of Feature Extraction in Single-Trial EEG Classification*, Journal of Neural Engineering, Volume 5, 2008. https://iopscience.iop.org/article/10.1088/2552/aab2f2#fnref-jneaab2f2bib082
- Sariq Khan, *Correlation-Based Feature Selection in a Data Science Project*, Medium, 2020. https://medium.com/@sariq16/correlation-based-feature-selection-in-a-data-science-project-3ca08d2af5c6