SEIZURE DETECTION USING POWER SPECTRAL DENSITY VIA HYPERDIMENSIONAL COMPUTING

Lulu Ge, and Keshab K. Parhi

University of Minnesota

ABSTRACT

Hyperdimensional (HD) computing holds promise for classifying two groups of data. This paper explores seizure detection from electroencephalogram (EEG) from subjects with epilepsy using HD computing based on power spectral density (PSD) features. Publicly available intra-cranial EEG (iEEG) data collected from 4 dogs and 8 human patients in the Kaggle seizure detection contest are used in this paper. This paper explores two methods for classification. First, few ranked PSD features from small number of channels from a prior classification are used in the context of HD classification. Second, all PSD features extracted from all channels are used as features for HD classification. It is shown that for about half the subjects small number features outperform all features in the context of HD classification, and for the other half, all features outperform small number of features. HD classification achieves above 95% accuracy for six of the 12 subjects, and between 85-95% accuracy for 4 subjects. For two subjects, the classification accuracy using HD computing is not as good as classical approaches such as support vector machine classifiers.

Index Terms— Hyperdimensional (HD) computing, power spectral density (PSD), and seizure detection.

1. INTRODUCTION

The hyperdimensional computing (HD) has its potential not only in addressing cognitive tasks [1, 2] such as analytical reasoning, but also in solving classification problems [3], e.g., language recognition, speech recognition, image classification, etc. In general, HD computing manipulates its unique datatype—*hypervectors*, which have their dimensionality d to be in the thousands, e.g., $d = 10,000$. Their components can be binary, integer, real or complex [4]; however, from a hardware-friendly perspective, this paper mainly focuses on binary hypervectors where the bit value is either 0 or 1.

This paper was supported in parts by NSF grant CCF-1814759 and by the Chinese Scholarship Council (CSC).

As an alternative for deep learning, HD computing is more hardware efficient in some applications such as speech recognition [5]. Power spectral density (PSD) features, relative band power and ratios of band power features have been used for seizure prediction [6–8] and for detection [9]. Moreover, it has been shown that small number of PSD features obtained from a feature ranking approach can achieve high accuracy using a support vector machine (SVM) classifier [9]. This paper explores seizure detection with HD computing using PSD features. The core of using HD computing lies in how to encode the PSD features and channel information by hypervectors, and how to generate *class* hypervectors. Therefore, three distinct encoding approaches are explored for seizure detection from the PSD features. These include: concatenating feature hypervectors to generate long class hypervectors, using multiple classifiers where each feature is used to train a single classifier and then the outputs of multiple classifiers are processed to generate the classification result, and training a single classifier using all feature hypervectors. All three approaches are applied to selected PSD features as well as all PSD features.

The remainder of this paper is organized as follows. Section 2 presents a review of HD computing and an overview of HD classification. Both small number of PSD features and all PSD features using HD computing are explored with three encoding approaches in Section 3, respectively. Finally, Section 4 concludes the paper.

2. PRELIMINARIES

2.1. IEEG Dataset

Table 1 lists the information of the Kaggle iEEG dataset [10]. Twelve subjects' iEEG recordings are analyzed for detecting seizures. The sampling frequency, f_s, of these recordings is 400 Hz for 4 dogs, and is 500 Hz or 5000 Hz for 8 human patients.

2.2. HD Computing

Performed among hypervectors, two basic mathematical operations—addition and multiplication—are used in

Table 1: Dataset Information

Patient	#ictal	#interictal	#test	#channel	f_s (Hz)
Dog_1	178	418	3181	16	400
Dog_2	172	1148	2997	16	400
Dog_3	480	4760	4450	16	400
Dog_4	257	2790	3013	16	400
Patient_1	70	104	2050	68	500
Patient_2	151	2990	3894	16	5000
Patient_3	327	714	1281	55	5000
Patient_4	20	190	543	72	5000
Patient_5	135	2610	2986	64	5000
Patient_6	225	2772	2997	30	5000
Patient_7	282	3239	3601	36	5000
Patient_8	180	1710	1922	16	5000

this paper for HD computing. The result of HD computing is measured by similarity. In terms of binary HD computing, the addition is typically associated with majority rule to ensure the output hypervector is binary. If the number of addends is even, then the tie is broken by adding an extra random hypervector to reduce the bias. The multiplication is exactly XOR operation and denoted as $\oplus$. Hamming distance is the similarity measurement metric as computed in Eq. (1), where d is the dimensionality of the hypervectors. For example, $\text{Ham}(\mathbf{A}, \mathbf{B}) = 0$ indicates the hypervectors $\mathbf{A}$ and $\mathbf{B}$ are identical, while $\text{Ham}(\mathbf{A}, \mathbf{B}) = 0.5$ means they are orthogonal or dissimilar. Thus, the closer the $\text{Ham}(\mathbf{A}, \mathbf{B})$ is to 0, the more similar are $\mathbf{A}$ and $\mathbf{B}$. Interested readers are referred to [3] for more details on HD computing and classification.

$$\text{Ham}(\mathbf{A}, \mathbf{B}) = \frac{1}{d} \sum_{i=1}^{d} 1_{\mathbf{A}(i) \neq \mathbf{B}(i)} \quad (1)$$

2.2.1. *Classification Overview with HD Computing*

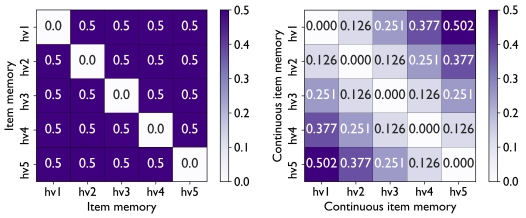
1). During learning phase, class hypervectors are trained and stored in the associative memory (AM) by encoding the training data with the hypervectors, which are prestored in the *item memory* (IM) or *continuous item memory* (CiM). More details for these two types of memories are described later. *2).* During inference phase, *query* hypervectors are generated based on the testing data in a similar encoding way, and then fed into AM to perform a similarity measurement with all pre-trained class hypervectors. The label indicated by the highest similarity is assigned as the final predicted result.

2.2.2. *IM and CiM*

Encoders may consider two types of hypervectors, which are generated in two different ways and then stored in IM and CiM, respectively.

IM: Statistics shows that the randomly generated hypervectors are nearly all dissimilar to each other. As shown in Fig. 1(a), the similarity is described in a heatmap, which indicates that five randomly generated hypervectors are orthogonal to each other.

CiM: Sometimes similarity should be retained when encoding the data. For example, to encode a feature vector, whose values are in a fixed range, hypervectors are assumed to preserve some correlation. In such a case, three steps are required: *1).* Quantize the given range into q levels; thus in total q *level* hypervectors are needed to be generated. *2).* Since adjacent values show greater correlation, the CiM starts with a randomly generated hypervector $\mathbf{L}_1$ with dimensionality d, which represents the smallest value of the quantized range, and is used to generate the final hypervector $\mathbf{L}_q$. Hypervector $\mathbf{L}_1$ is dissimilar with $\mathbf{L}_q$, which means half the bits are different. To realize this, we can randomly choose $d/2$ components of $\mathbf{L}_1$ and then split them into $(q-1)$ groups. The other hypervectors are generated from $\mathbf{L}_1$ by flipping components from one group to another. Similarity among five hypervectors in CiM is displayed in Fig. 1(b), which reflects adjacent hypervectors show more correlations.



(a) Item memory. (b) Continuous item memory.

Fig. 1: Similarity among hypervectors in IM and CiM.

3. PSD METHOD

Based on [9], PSD achieves high classification accuracy on the Kaggle contest dataset using classification and regression tree (CART) based feature selection and polynomial SVM classifier. In this paper, we use HD computing with selected small number of PSD features as well as using all PSD features.

All the iEEG data are preprocessed to extract band powers and remove power line noise [9]. *1).* For dog subjects, the frequency band is split into 10 frequency sub-bands (Hz): 3-8, 8-13, 13-30, 30-55, 55-80, 80-105, 105-130, 130-150, 150-170, 170-200. *2).* For human subjects, the frequency band is split into 13 frequency sub-bands (Hz): 3-8, 8-13, 13-30, 30-50, 50-80, 80-100, 100-130, 130-160, 160-200, 200-250, 250-300, 300-350, 350-400. To eliminate power line hums at 60 Hz and its harmonics, spectral powers in the band of $[60i–3, 60i+3]$ Hz are excluded in the PSD computation, where $i \in [0, 6]$.

Three spectral power metrics are computed as shown in Eq. (3), where ASP_{f_1,f_2} refers to the absolute spectral power of a signal in the frequency band $[f_1, f_2]$ Hz, RSP_{f_1,f_2} refers to the relative spectral power in band $[f_1, f_2]$ Hz and $\text{Ratio}_{f_1,f_2,f_3,f_4}$ represents the spectral power ratio of the absolute spectral power in band $[f_1, f_2]$ Hz over that in band $[f_3, f_4]$ Hz.

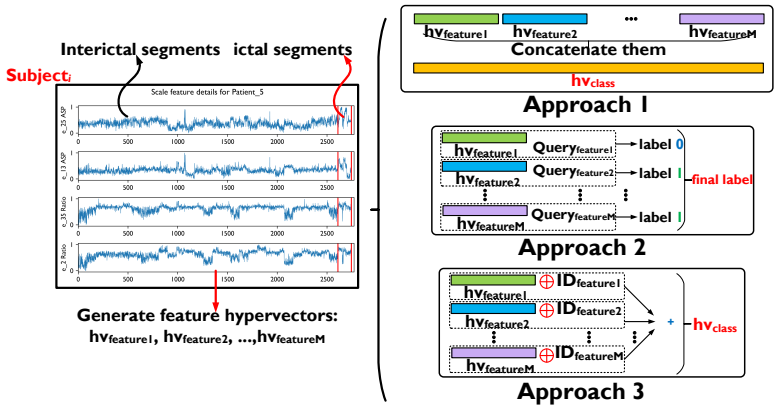


Fig. 2: Diagram for PSD method with HD computing.

$$\mathbf{hv_{clip_i}} = \left[\left[\mathbf{ID_1} \oplus \bar{\mathbf{L}}_\mathbf{1} + \cdots + \mathbf{ID_M} \oplus \bar{\mathbf{L}}_\mathbf{M}\right] \oplus \mathbf{Ch_1} + \cdots + \left[\mathbf{ID_1} \oplus \bar{\mathbf{L}}_\mathbf{1} + \cdots + \mathbf{ID_M} \oplus \bar{\mathbf{L}}_\mathbf{M}\right] \oplus \mathbf{Ch_K}\right] \quad (2a)$$

$$\mathbf{hv_{class}} = \left[\mathbf{hv_{clip_1}} + \mathbf{hv_{clip_2}} + \cdots + \mathbf{hv_{clip_N}}\right] \quad (2b)$$

$$\mathrm{ASP}_{f_1,f_2} = \log \textstyle\sum_{f\in[f_1,f_2]} \mathrm{PSD}(f) \quad (3a)$$

$$\mathrm{RSP}_{f_1,f_2} = \log \frac{\sum_{f\in[f_1,f_2]} \mathrm{PSD}(f)}{\sum_{\text{all } f} \mathrm{PSD}(f)} \quad (3b)$$

$$\mathrm{Ratio}_{f_1,f_2,f_3,f_4} = \mathrm{ASP}_{f_1,f_2} - \mathrm{ASP}_{f_3,f_4} \quad (3c)$$

3.1. Small Number of PSD Features

This paper used the same PSD features selected in [9] using CART (see Table I of [9]). These features are scaled into the range $[0, 1]$. This scaling step facilitates the encoding process for HD computing. Now the problem is how to generate a *class* hypervector $\mathbf{hv_{class}}$ based on N clips with M PSD features.

3.1.1. Approach 1

Before training, quantize the range $[0, 1]$ into q levels, then for each subject, in total q level hypervectors $\{\mathbf{L_1}, \mathbf{L_2}, \cdots, \mathbf{L_q}\}$ are generated in the CiM. As shown in Fig. 2, clip i corresponds to a data point and then should be represented by a *clip* hypervector $\mathbf{hv_{clip_i}}$ based on its quantized level, where $\mathbf{hv_{clip_i}} \in \{\mathbf{L_1}, \mathbf{L_2}, \cdots, \mathbf{L_q}\}$ and $i \in [1, N]$. As shown in Eq. (4), In total M *feature* hypervectors $\mathbf{hv_{feature_j}}$ should be trained by adding all corresponding clip hypervectors, where $j \in [1, M]$. The final class hypervector is generated by concatenating all feature hypervectors. For example, Dog_1 requires 3 features as [9]. Since each feature hypervector $\mathbf{hv_{feature_j}}$ has its dimensionality $d = 10,000$, the dimensionality for $\mathbf{hv_{class}}$ becomes $30,000$.

$$\mathbf{hv_{feature_j}} = \left[\mathbf{hv_{clip_1}} + \cdots + \mathbf{hv_{clip_N}}\right], \quad (4a)$$

$$\mathbf{hv_{class}} = (\mathbf{hv_{feature_1}}, \cdots, \mathbf{hv_{feature_M}}). \quad (4b)$$

3.1.2. Approach 2

Approach 2 uses M feature hypervectors $\mathbf{hv_{feature_j}}$ to represent a class, where $j \in [1, M]$. This is different from Approach 1 which employs a single class hypervector $\mathbf{hv_{class}}$. Therefore, a given test segment will produce M query hypervectors. Similarity measurement should be performed for each query hypervector. We obtain the label results from M classifiers. The final label is determined by a majority vote of all label results. For example, Patient_3 needs 4 features to determine the label. In this case, 4 feature hypervectors are trained. If the label 0 for interictal class is assigned 3 times, and 1 for ictal class is assigned once, then the final label should be classified as 0, namely the interictal segment. If the number of labels is even and half the labels correspond to each class, the tie is broken in favor of detection.

3.1.3. Approach 3

Instead of concatenating the feature hypervectors, the hypervector $\mathbf{hv_{class}}$ in Approach 3 is generated by Eq. (5), where the hypervectors' index ($\mathbf{ID}$) values are pregenerated in IM, whose total number is the same as selective features.

$$\mathbf{hv_{clip_i}} \in \{\mathbf{L_1}, \mathbf{L_2}, \cdots, \mathbf{L_q}\}, \text{ where } i \in [1, N], \quad (5a)$$

$$\mathbf{hv_{feature_j}} = \left[\mathbf{hv_{clip_1}} + \cdots + \mathbf{hv_{clip_N}}\right], \quad (5b)$$

$$\mathbf{hv_{class}} = \left[\mathbf{hv_{feature_1}} \oplus \mathbf{ID_1} + \cdots + \mathbf{hv_{feature_M}} \oplus \mathbf{ID_M}\right] \quad (5c)$$

3.2. All PSD Features

We also take all features into consideration, which means all the three spectral power metrics over all sub-bands are computed. Therefore, *1).* for dog subjects who have

Table 2: Simulation Results for Selective Features with Quantization Level $q = 21$

Patient	Approach 1		Approach 2		Approach 3	
	A_{train} (%)	A_{test} (%)	A_{train} (%)	A_{test} (%)	A_{train} (%)	A_{test} (%)
Dog_1	97.9866	**99.2455**	97.3154	99.1198	97.3154	97.9881
Dog_2	94.7727	**95.8625**	94.0909	93.7604	92.8788	95.4621
Dog_3	96.8511	96.1124	97.3282	**96.2247**	96.9084	96.2022
Dog_4	79.0942	**86.2264**	77.1250	85.3966	86.1831	83.0070
Patient_1	93.6782	**95.7561**	74.1379	87.4634	92.5287	95.7561
Patient_2	85.1640	23.9086	66.6985	12.9173	78.4782	34.5146
Patient_3	75.0240	**94.0671**	76.4649	87.0414	73.5831	85.0898
Patient_4	71.4286	72.9282	69.5238	70.5341	77.6190	75.3223
Patient_5	74.6812	84.4943	83.6066	92.1969	87.9781	**93.1681**
Patient_6	88.4885	86.6200	84.4845	83.4168	91.3580	89.8565
Patient_7	93.5246	84.6709	55.8364	30.0750	87.5604	80.3388
Patient_8	66.7196	71.0198	9.5238	9.3652	61.6931	67.6899

Table 3: Simulation Results for All Features with Quantization Level $q = 21$

Patient	Approach 1		Approach 2		Approach 3	
	A_{train} (%)	A_{test} (%)	A_{train} (%)	A_{test} (%)	A_{train} (%)	A_{test} (%)
Dog_1	74.8322	36.8752	72.1477	22.1000	95.8054	59.7925
Dog_2	83.4848	75.0417	75.1515	72.0387	93.5606	75.4087
Dog_3	92.1183	94.4944	78.2252	92.4270	95.8206	94.6067
Dog_4	76.5671	42.1839	66.7542	36.4089	96.0289	45.8679
Patient_1	67.2414	83.7073	66.6667	14.0488	100.0000	66.3902
Patient_2	78.4464	45.1464	76.9500	63.9188	96.3069	73.3950
Patient_3	79.8271	85.2459	78.0019	58.7041	98.7512	89.6956
Patient_4	75.2381	85.4512	74.2857	85.4512	95.7143	**88.9503**
Patient_5	90.6011	87.2739	76.6120	72.3041	94.9362	92.1299
Patient_6	98.4985	91.9253	97.5309	91.3247	99.3660	**92.4258**
Patient_7	97.0747	**88.6143**	42.6015	38.6282	98.8640	82.1716
Patient_8	83.9683	73.7773	78.0423	63.9438	89.7884	**77.8876**

10 sub-bands, a total of $65(= 10 + 10 + \binom{10}{2})$ features need to be computed. They are 10 ASP, 10 RSP and $\binom{10}{2}$ Ratio. *2).* Similarly, for human subjects with 13 sub-bands, $104(= 13 + 13 + \binom{13}{2})$ features are computed.

Similar to small number of PSD features method, three approaches are performed for all features with HD computing. The only difference is the generation of $\mathbf{hv_{class}}$ hypervector in Approach 3, which is shown in Eq. (2). The algorithm is straightforward: *1).* Before training, pre-generate K channel hypervectors $\mathbf{Ch_k}$ in IM_1, M ID hypervectors $\mathbf{ID_j}$ in IM_2, and q level hypervectors $\{\mathbf{L_1}, \mathbf{L_2}, \cdots, \mathbf{L_q}\}$ in CiM, where $k \in [1, K]$ and $j \in [1, M]$. *2).* For each clip, generate the clip hypervector as described in Eq. (2a), where $[\cdot]$ represents the majority rule and $\bar{\mathbf{L}}_\mathbf{j} \in \{\mathbf{L_1}, \cdots, \mathbf{L_q}\}$. The final class hypervector is generated by adding all clip hypervectors for the same class as shown in Eq. (2b). *3).* Once the two class hypervectors are trained, during the testing phase, the query hypervector is generated by each clip in the same way as shown in Eq. (2a). The label is assigned according to the similarity measurement between the query hypervetors and the trained two class hypervetors.

Simulation results for the PSD method with small number of PSD features and all PSD features are shown in Tables 2 and 3, respectively. In Table 2, the results for Patient_2 is are poor for all approaches. Compared to small number of features, in terms of test accuracy shown in Table 3, the test accuracy of Patient_2 has been improved. However, some cases like Dog_1 and Dog_4 get worse.

4. DISCUSSION AND CONCLUSION

This paper combines PSD features with HD computing for detecting seizures. Both small number of PSD features and all PSD features are studied. For about half of the subjects, small number features outperform all features in the context of HD classification, and for the other half, all features outperform small number of features. HD classification achieves above 95% accuracy for six of the 12 subjects, and between 85-95% accuracy for 4 subjects. For two subjects, the classification accuracy using HD computing is not as good as classical approaches such as SVM [9]. Future efforts will address developing feature ranking methods for HD classification. The proposed PSD approach should be compared with other approaches such as the local binary pattern (LBP) [11].

References

[1] Pentti Kanerva, *Sparse Distributed Memory*, MIT press, 1988.

[2] Pentti Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, no. 2, pp. 139–159, 2009.

[3] Lulu Ge and Keshab K Parhi, "Classification using Hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[4] Abbas Rahimi, Pentti Kanerva, Luca Benini, and Jan M Rabaey, "Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of ExG signals," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 123–143, 2018.

[5] Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing, "VoiceHD: Hyperdimensional computing for efficient speech recognition," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–8.

[6] Zisheng Zhang and Keshab K Parhi, "Low-complexity seizure prediction from iEEG/sEEG using spectral power and ratios of spectral power," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 3, pp. 693–706, 2016.

[7] Keshab K Parhi and Zisheng Zhang, "Discriminative ratio of spectral power and relative power features derived via frequency-domain model ratio with application to seizure prediction," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 4, pp. 645–657, 2019.

[8] Yun Park, Lan Luo, Keshab K Parhi, and Theoden Netoff, "Seizure prediction with spectral power of EEG using cost-sensitive support vector machines," *Epilepsia*, vol. 52, no. 10, pp. 1761–1770, 2011.

[9] Zisheng Zhang and Keshab K Parhi, "Seizure detection using regression tree based feature selection and polynomial SVM classification," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 6578–6581.

[10] "Upenn and mayo clinic's seizure detection challenge," `https://www.kaggle.com/c/seizure-detection/data`.

[11] Alessio Burrello, Kaspar Anton Schindler, Luca Benini, and Abbas Rahimi, "Hyperdimensional computing with local binary patterns: One-shot learning for seizure onset detection and identification of ictogenic brain regions from short-time ieeg recordings.," *IEEE transactions on bio-medical engineering*, vol. 67, no. 2, pp. 601–613, 2020.