Learning Multiscale Neural Metrics via Entropy Minimization

Austin J. Brockmeier, Luis G. Sanchez Giraldo, John S. Choi, Joseph T. Francis, and Jose C. Principe

***Abstract*— In order to judiciously compare neural responses between repeated trials or stimuli, a well-suited distance metric is necessary. With multi-electrode recordings, a neural response is a spatiotemporal pattern, but not all of the dimensions of space and time should be treated equally. In order to understand which dimensions of the input are more discriminative and to improve the classification performance, we propose a metric-learning approach that can be used across scales. This extends previous work that used a linear projection into lower dimensional space; here, multiscale metrics or kernels are learned as the weighted combinations of different metrics or kernels on each of the neural response's dimensions. Preliminary results are explored on a cortical recording of a rat during a tactile stimulation experiment. Metrics on both local field potential and spiking data are explored. The learned weights reveal important dimensions of the response, and the learned metrics improve nearest-neighbor classification performance.**

I. INTRODUCTION

Evaluating the differences between neural responses to different stimuli or conditions is a fundamental but surprisingly difficult problem. Outside of training statistical models or psychophysical evaluations, a natural measure of dissimilarity between neural responses is lacking. Should evoked responses of local field potentials (LFPs) be compared via Euclidean distance or cross-correlation? How much information is lost by binning spike trains? Fortunately, this is a well-investigated problem as finding a 'good' similarity measure is a common problem in pattern recognition and machine learning.

The approach we consider is metric-learning for classification [1], [2]. The concept of metric-learning is to parametrize a distance function such that examples from the same class are deemed close and examples from different classes are considered far apart. Here training set classification error is not used, instead the metric is learned via an information-theoretic optimization problem using the class label information [3]. A nearest-neighbor assignment is performed post-hoc to evaluate the learned metric.

As opposed to previous work [4], we avoid using linear projections, instead learning new metrics/kernels as the combination metrics/kernels for different dimensions of the neural response. This enables us to learn multiscale measures such as combinations of spike-train metrics or weighted combinations of kernels with different kernel sizes. Specifically, we explore using tensor product and direct sum kernels on both LFPs and spike trains.

This work was supported in part by the Univ. Florida Graduate School Fellowship and DARPA Contract N66001-10-C-2008.

A. J. Brockmeier, L. G. Sanchez Giraldo, and J. C. Príncipe are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: {ajbrockmeier,sanchez,principe}@cnel.ufl.edu)

J. S. Choi and J. T. Francis are with the Department of Physiology and Pharmacology, State University of New York Downstate Medical School, Brooklyn, NY 11203 USA.

II. METHOD

A. Neural data representations

Modern multi-electrode arrays record a combination of local field potentials (LFPs) and action potentials (spikes) from nearby neurons. The spike timing extracted from the voltage time-series can be used directly or binned into an instantaneous firing rate using non-overlapping fixed-width bins.

For the classification of different trial conditions, the sample from each trial is the concatenation of all selected channels or neurons. We learn parameters for each of these dimensions across different scales: a temporal weighting for LFPs, a combined spatiotemporal weighting of binned spike trains, and a unit-wise weighting of a spike train metric [5], [6].

We consider the N-dimensional neural response to a given trial as $x = [x_{(1)} \dots x_{(N)}]$, where parenthetical subscripts denote the response dimension, corresponding to either a different time lag, channel, unit, or combined spatiotemporal index. Let x_j denote the neural response for the jth trial, $j \in \{1, \dots, n\}$, and let $l_j \in \{1, \dots, m\}$ denote the discrete class label corresponding to a certain condition or stimulus for the jth trial.

B. Metrics and Kernel Representation

To form a multiscale metric of the neural response, we first begin with a metric for each dimension of the neural response $d(x_{(1)}, x'_{(1)}), \dots, d(x_{(N)}, x'_{(N)})$. Based on the metric, a measure of similarity between neural responses can be formed by applying the Gaussian kernel, but in order for the similarity to be positive definite, the metric in the argument of the exponential should be Euclidean. (If it is not, any distance matrix or even dissimilarity matrix can be transformed to meet this requirement [7].) The similarity between a pair of samples x and x' on the ith dimension is $\kappa(x_{(i)}, x'_{(i)}) = \exp(-\theta_i d^2(x_{(i)}, x'_{(i)}))$, where θ_i is a kernel size parameter. Changing the kernel size adjusts how close samples must be in order to be considered similar.

In terms of a group of samples, the pairwise squared distance matrix for the ith dimension is denoted $D_i^{\cdot 2}$ where $[D_i^{\cdot 2}]_{j,k} = d^2(x_j(i), x_k(i))$ $j, k \in \{1, \dots, n\}$. Likewise, the corresponding kernel matrix with kernel size parameter θ is $K_i = \exp(-\theta D_i^{\cdot 2})$. (The notation $D^{\cdot 2}$ denotes that each element is squared, opposed to $D^2 = DD$.)

Considering two input dimensions at once, a joint similarity measure between a pair of samples $x_{(1,2)} = [x_{(1)}, x_{(2)}]$ and $x'_{(1,2)} = [x'_{(1)}, x'_{(2)}]$ is formed from the tensor product between the two kernels, denoted $\otimes$ is

$$(\kappa_1 \otimes \kappa_2)(x_{(1,2)}, x'_{(1,2)}) = \kappa_1(x_{(1)}, x'_{(1)}) \cdot \kappa_2(x_{(2)}, x'_{(2)}).$$

In terms of the kernel matrices, this corresponds to the element-wise product, the Hadamard product, between the kernel matrices $K_1 \circ K_2$.

The labels of the trials can also be represented by a kernel matrix L, where each entry $L_{j,k} = \delta(l_j, l_k)$ uses the 0-1 kernel

$$\delta(l, l') = \begin{cases} 1 & \text{if } l = l' \\ 0 & \text{if } l \neq l' \end{cases} . \tag{1}$$

C. Entropy

Rényi's α-order entropy is an information measure for probability distributions. Let $[p_1 \dots p_N]$ denote the probability mass function (pmf) for a discrete random variable X; then the entropy is evaluated as $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p_i^\alpha$. For $\alpha \to 1$, the measure approaches Shannon's entropy $\sum_i -p_i \log p_i$. Interestingly, a similar quantity can be defined in terms of the eigenvalues of a positive definite kernel matrix [3].

For any positive definite matrix A, the trace is equal to the sum of the eigenvalues $\mathrm{tr}(A) = \sum_i \lambda_i(A)$. Let $B = \frac{A}{\mathrm{tr}(A)}$; B is also positive definite, and by normalizing by the trace, the eigenvalues add to 1. This means all functions of pmfs will have the same properties when applied to the eigenvalues of B. Using the formulation of Rényi's entropy on the eigenvalues yields a matrix-based analog to entropy [3]:

$$S_\alpha(A) = \frac{1}{1-\alpha} \log \sum_i \lambda_i^\alpha(B), \tag{2}$$

where $\mathrm{tr}(B^\alpha) = \sum_i \lambda_i^\alpha(B)$.

Unlike parametric approaches that require knowledge of the distributions or non-parametric approaches that estimate the density using methods such as Parzen windows, this approach directly estimates an entropy quantity without ever requiring an explicit density function. Another key benefit is that optimization can be done in terms of the kernel and distance matrix using matrix calculus.

For discrete symbols, the plug-in estimate of the entropy is equal to the proposed matrix-based analog $S_\alpha(L) = 1/(1-\alpha) \log \sum_i \hat{p}_i^\alpha$, when using the 0-1 kernel $\hat{p}_i = 1/n \sum_{j=1}^n \delta(i, l_j)$.

D. Joint Entropy

The entropy of the joint measure is calculated as the entropy of trace-normalized kernel matrices B and C after re-normalization [3]:

$$S_\alpha(B, C) = \frac{1}{1-\alpha} \log \left[\mathrm{tr} \left(\left(\frac{B \circ C}{\mathrm{tr}(B \circ C)} \right)^\alpha \right) \right]. \tag{3}$$

From this a measure of conditional entropy $S_\alpha(B|C) = S_\alpha(B, C) - S_\alpha(C)$ can be applied, and this form of conditional entropy was used in previous work for learning a Mahalanobis distance [3], [4].

E. Tensor Product Kernel

As mentioned above, the tensor product between kernel functions is a joint measure for multivariate data formed as the product of kernels for each dimension of the input.

The tensor product between Gaussian kernels corresponds to using a nonnegative combination of the different Euclidean metrics as the argument to the kernel function $\kappa_\theta(x, x') = \exp(-\sum_i \theta_i d^2(x_{(i)}, x'_{(i)}))$, and $K_\theta = \exp(-\sum_i \theta_i D_i^2)$. Adjusting the parameters of $\theta \geq 0$ changes the kernel size of each component kernel, and this is equivalent to scaling each dimension of the input so as to form a new metric $d_\theta^2(x, x') = \sum_i \theta_i d^2(x_{(i)}, x'_{(i)}) = \sum_i d^2(\sqrt{\theta_i} x_{(i)}, \sqrt{\theta_i} x'_{(i)})$.

In order to learn this metric we consider the following information-theoretic optimization problem:

$$\begin{aligned} \underset{\theta \geq 0}{\text{minimize}} \;\; & S_\alpha(L, K_\theta) \\ \text{subject to} \;\; & S_\alpha(K_\theta) = \eta \end{aligned} \tag{4}$$

This problem attempts to minimize the joint entropy of the data and the labels while constraining the marginal entropy. A natural choice for η is close to the Rényi's entropy of the class labels, which for labels equally distributed among m classes is $\log(m)$.

The relationship with the conditional entropy ($S_\alpha(L|K) = S_\alpha(L, K) - S_\alpha(K)$) can be seen by transforming the constraint into the Lagrangian formulation

$$\underset{\theta \geq 0}{\text{minimize}} \;\; S_\alpha(L, K_\theta) - \lambda \left(S_\alpha(K) - \eta \right) \tag{5}$$

with the Lagrange multiplier λ. However, the constraint on the non-negativity of the coefficients remains. As a simple approach, an unconstrained optimization is formed by re-parametrizing the function in terms of w where $\theta_i = 10^{w_i}$. By solving this optimization problem, a new metric can be learned as a linear combination of metrics.

F. Direct Sum Kernel

As an alternative formulation, we consider composing a kernel as a nonnegative combination of positive definite kernels $K_\theta = \sum_j \theta_j K_j$. For nonnegative combinations $\theta_j \geq 0$, the new kernel K_θ corresponds to a positive definite similarity function that can be embedded in Euclidean space.

Each dimension of the input may have one or many kernels associated with it, each with a different shape or kernel size. The parameters of component kernels are not adapted and must be chosen a priori.

In order to learn the weights of the sum kernel, a modification of the original conditional entropy metric-learning optimization [3] is proposed:

$$\begin{aligned} \underset{\theta \geq 0}{\text{minimize}} \;\; & S_\alpha(L|K_\theta) \\ \text{subject to} \;\; & \textstyle\sum_j \theta_j = 1 \\ & \forall j \;\; \theta_j \geq 0 \end{aligned} \tag{6}$$

where $S_\alpha(L|K_\theta) = S_\alpha(L, K_\theta) - S_\alpha(K_\theta)$. The convex constraint $\sum_j \theta_j = 1$ prevents trivial solutions.

The constraints can be enforced by using a parameterization in terms of another variable w where $\theta_j = \frac{\exp(w_j/\tau)}{\sum_j \exp(w_j/\tau)}$, where τ is the temperature parameter. This softmax activation function enforces positivity and forces the coefficients to add to 1. Gradient descent can still be used for optimizing w, and after every step the softmax function is applied to enforce these constraints. (Typically, the temperature parameter is annealed during learning; here it was fixed at $\tau = 1$.)

G. Derivatives

Consider the eigendecomposition of the kernel matrix $B = U\Lambda U^{\mathrm{T}}$. The matrix function $B^\alpha = U\Lambda^\alpha U^{\mathrm{T}}$ changes the eigenvalues, but the eigenvectors stay the same, and the gradient is $\nabla B^\alpha = \alpha U \Lambda^{\alpha-1} U^{\mathrm{T}}$. Using this, the gradient of $S_\alpha(B)$ follows as

$$\nabla S_\alpha(B) = \frac{\alpha}{(1-\alpha)\mathrm{tr}(B^\alpha)} U\Lambda^{\alpha-1}U^{\mathrm{T}}. \tag{7}$$

The gradient with respect to the joint entropy involves the matrix calculus identity $\nabla(B \circ C) = B \circ (\nabla C) + (\nabla B) \circ C$.

For the product kernel, the gradient of the Gaussian kernel matrix with respect to the kernel size is $\frac{\partial K_\theta}{\partial \theta_i} = -K_\theta \circ D_i^2$. Using these properties, the gradient for learning the parameters of the product kernel is

$$\frac{\partial S_\alpha(L, K_\theta)}{\partial \theta_i} = \sum_{j,k} \left(-D_i^2 \circ K_\theta \circ \nabla S_\alpha(L, K_\theta)\right)_{jk}, \tag{8}$$

and for the sum kernel it is

$$\frac{\partial S_\alpha(L|K_\theta)}{\partial \theta_i} = \sum_{j,k} \left(K_i \circ (\nabla S_\alpha(L, K_\theta) - \nabla S_\alpha(K_\theta))\right)_{jk}. \tag{9}$$

III. Neural Recording: Cortical Response to Rat Forepaw Taction

All animal procedures were approved by the SUNY Downstate Medical Center IACUC and conformed to National Institutes of Health guidelines.

Cortical LFPs and spikes were recorded during natural tactile stimulation (light thwacks of forepaw digits and palm) of a female Long-Evans rat under anesthesia. After administering isofluorane, a 32-channel Michigan Probes electrode array was inserted into the hand region of primary somatosensory cortex (S1). The array had 8 contacts on each of 4 shanks. Another array was inserted into VPL region of the thalamus, but the signals are not used here.

Using a motorized probe, a rat's forepaw was touched 225 times at 9 sites (4 digits and 5 sites on the palm). For each touch site, the probe was positioned 4mm above the surface of the skin and momentarily pressed down for 150ms; this was repeated 25 times at random intervals. For analysis, only a 170ms window following the touch onset was used.

Automatic spike-sorting selected 38 multi-neuron units from the 32 channels. Of these only 24 units were used whose average firing rate was below 30Hz in the 170ms after touch. The LFPs were filtered with cutoffs (5Hz, 300Hz) and sampled at a rate of 1220.7Hz. The signals were digitally filtered using a 3rd-order Butterworth high-pass filter with cutoff of 4Hz and notch filters at 60Hz and its first 5 harmonics. The first 170ms after touch corresponded to 208 discrete time samples.

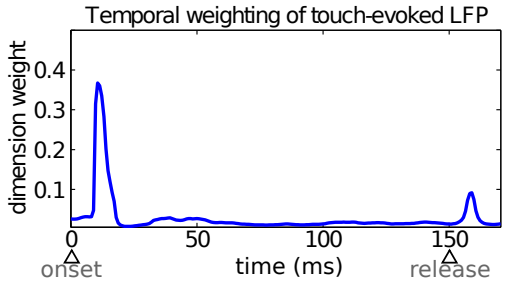

Fig. 1. The temporal weighting of the cortical LFP response (32 channels in S1) learned to distinguish the touch site. The two peaks follow touch onset and release. This weighting was learned for the product kernel. On this run the classification accuracy increased from 54.17% to 70.83%.

TABLE I

NN classification using LFPs. Metric-learning is able to increase the classification rate by over 10% using the product kernel and over 25% using the sum kernel.

	mean	S.D.
Original sum	43.54%	5.12%
Original product	54.51%	4.55%
Learned sum	79.79%	4.29%
Learned product ($\eta = 3$)	66.32%	5.39%
Learned product ($\eta = 1$)	64.37%	4.04%

IV. Results

We explored the metric-learning approach on the tactile dataset. As there were 9 classes with equal number of classes, we chose $\eta \in \{1, 3\}$ since $\log(9) \approx 2.2$. For all experiments, we used gradient descent with a fixed stepsize of 0.5, and the order of Rényi's entropy was fixed at $\alpha = 1.01$.

A. Metric Learning for LFPs

The LFP response to each touch was a 208×32 spatiotemporal pattern. We treated the 32-dimensional vector at each time lag as its own input dimension, with the goal of learning a temporal weighting to discriminate the touch sites.

The Euclidean distance in $\mathbb{R}^{32}$ was used for each time lag. For the product kernel, the goal is to learn a weighted combination of these 208 distances. One example of a learned weighting is shown in Fig. 1. From the weighting it is clear that the multi-channel LFP response 15ms after both touch onset and release is most important to discriminate between the touch sites.

For the sum kernel, Euclidean distance was computed for each input dimension. For each of these distance matrices, the standard Gaussian kernel $\kappa(d) = \exp(-.5d^2/\sigma^2)$ was applied with 4 different kernel sizes $\sigma \in [.1, 1, 10, 100]$, in total $4 \cdot 208 = 832$ weights were learned. The softmax activation function with $\tau = 1$ encourages sparseness, and after training only a single weight was non-zero; the non-zero weight corresponded to the time bin 15ms after touch onset, at the same peak found with the product kernel see Fig. 1. The resulting metric was the highest performing with average

nearest-neighbor (NN) classification of nearly 80%. Thus, at the peak of the response the spatial amplitude vector is very discriminative. For both approaches, the nearest-neighbor classification results, across 20 Monte Carlo divisions of the dataset into 2/3 for training and 1/3 for testing, are shown in Table I.

B. Metric Learning for Spikes

We apply metric-learning to spike trains using two different representations: binning and treating the resulting vectors in Euclidean space, and using the Victor-Purpura spike-train metric. In both cases, learning a product kernel was able to increase the classification rate. However, the sum kernel failed to increase classification performance for both cases (data not shown).

1) Binned Spike Metric: The responses of 24 units in S1 were binned using non-overlapping windows of size 25ms or 50ms. (Other binsizes ranging from 5ms to 100ms were tested, but had worse performance and are not reported due to lack of space.) Each dimension of the combined spatiotemporal response was treated independently using the Euclidean metric. Using the product kernel, a weighted combination of the distances across all spatiotemporal dimensions was learned. The peristimulus time histograms for each touch site are shown in Fig. 2. The classification results for the 50ms binning are tabulated in Table II.

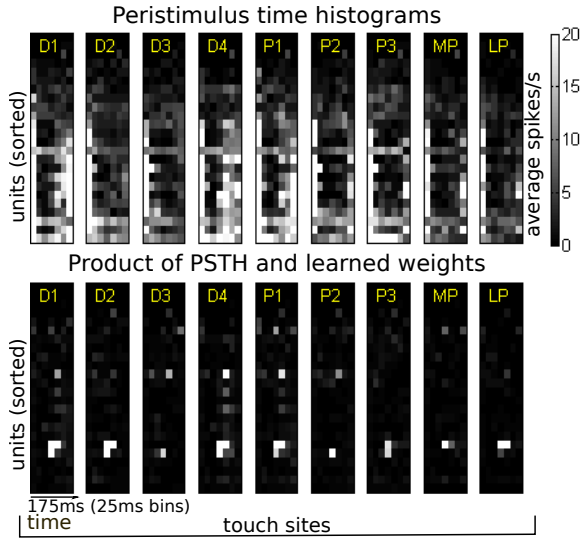

Fig. 2. (Top) The peristimulus time averages for the 9 touch sites: the responses of 24 spiking units in S1 following touch were binned (binwidth of 25ms) and averaged across trials. (Bottom) The product of the PSTHs and the learned spatiotemporal weighting. The weighting is sensitive to discriminative bins in the center of the response time, since the bins close to onset and release have large counts irrespective of the touch site. On this run the classification performance increased from 16.7% to 23.6%.

2) Spike Train Metric: Victor-Purpura distance [5] (VP distance) is a spike alignment metric. The distance between two spike trains is the minimum cost of moving, deleting, and adding individual spikes to either spike train to make them the same. The relative cost of moving versus adding a new spike is controlled by the temporal precision value q with units of s^{-1}. A Euclidean version of the VP distance [6] was used for each of the 24 units across 3 different temporal precision values of $(10, 100, 1000)s^{-1}$. A weighted combination of the 72 metrics was learned as a product kernel. The classification results are tabulated in Table II, with a negligible increase of the classification rate versus the binned approach.

TABLE II

NN CLASSIFICATION USING SPIKE TRAINS. METRIC-LEARNING FOR THE PRODUCT KERNEL IS ABLE TO INCREASE THE CLASSIFICATION RATE BY OVER 5% FOR BOTH THE BINNED REPRESENTATION AND THE L_2 VERSION OF VICTOR-PURPURA DISTANCE [6].

	L_2 binned at 50ms		L_2 VP distance	
	mean	S.D.	mean	S.D.
Original product	21.88%	3.92%	23.40%	3.97%
Learned product ($\eta = 3$)	27.64%	4.39%	28.33%	4.56%
Learned product ($\eta = 1$)	25.35%	4.46%	28.40%	4.93%

V. CONCLUSION

We presented an approach to learn weighted combinations of metrics or kernels for neural response classification, extending previous work [4]. The metric-learning hinges on a kernel matrix-based measure of entropy [3], [8].

Results on learning a temporal weighting for LFPs, a spatiotemporal weighting for multi-unit firing rates, and a spike-train metric sensitive to multiple timescales are all explored on a sensory processing dataset. For this particular dataset, the LFPs carry much more information about the stimulus than the recorded spike trains. This could be caused by using multi-neuron instead of single-neuron spike trains.

One of the benefits of metric learning is that the learned metric can be used to distinguish classes not seen during training. In future work, we would like to see how the learned metric is able to distinguish novel conditions.

REFERENCES

[1] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15*, vol. 15, 2002, pp. 505–512.

[2] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[3] L. G. Sanchez Giraldo and J. C. Principe, "Information theoretic learning with infinitely divisible kernels," in *International Conference on Learning Representations*, May 2013.

[4] A. J. Brockmeier, L. G. Sanchez Giraldo, M. S. Emigh, J. Bae, J. S. Choi, J. T. Francis, and J. C. Principe, "Information-theoretic metric learning: 2D linear projections of neural data for visualization," in *Engineering in Medicine and Biology Society (EMBC), 2013 Annual International Conference of the IEEE*, July 2013.

[5] J. D. Victor and K. P. Purpura, "Nature and precision of temporal coding in visual cortex: a metric-space analysis," *Journal of Neurophysiology*, vol. 76, no. 2, pp. 1310–1326, 1996.

[6] A. J. Dubbs, B. A. Seiler, and M. O. Magnasco, "A fast $\mathcal{L}_p$ spike alignment metric," *Neural Computation*, vol. 22, no. 11, pp. 2785–2808, 2010.

[7] J. C. Gower and P. Legendre, "Metric and Euclidean properties of dissimilarity coefficients," *Journal of classification*, vol. 3, no. 1, pp. 5–48, 1986.

[8] L. G. Sanchez Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," `arXiv:1211.2459 [cs.LG]`, November 2012.