ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9415011

TRAINING A BANK OF WIENER MODELS WITH A NOVEL QUADRATIC MUTUAL INFORMATION COST FUNCTION

Bo Hu and Jose C. Principe (Life Fellow, IEEE)

Dept. of Electrical and Computer Engineering, University of Florida

ABSTRACT

This paper presents a novel training methodology to adapt parameters of a *bank of Wiener models* (BWMs), i.e., a bank of linear filters followed by a static memoryless nonlinearity, using full *pdf* information of the projected outputs and the desired signal. BWMs also share the same architecture with the first layer of a *time-delay neural networks* (TDNN) with a single hidden layer, which is often trained with backpropagation. To optimize BWMs, we develop a novel cost function called the *empirical embedding of quadratic mutual information* (E-QMI) that is metric-driven and efficient in characterizing the statistical dependency. We demonstrate experimentally that by applying this cost function to the proposed model, our method is comparable with state-of-the-art neural network architectures for regressions tasks without using backpropagation of the error.

Index Terms— Wiener models, MIMO, information-theoretic learning, empirical embedding, regression

1. INTRODUCTION

The Wiener model belongs to a class of block-oriented models widely used in system identification for its simplicity of having a linear dynamic block (FIR filter) followed by a static nonlinearity. It is parameterized by a set of weights in the linear function, in spite of creating a nonlinear transfer function [1].

The parameter estimation of Wiener models is usually done by minimizing an error measurement with *mean square error* (MSE) or by maximizing the likelihood (ML) in a Bayesian setting [2, 3].

However, a Wiener model remains a *multiple-input single-output* (MISO) system, which means it only creates an one-dimensional projection space. To go beyond this limitation, we propose a new architecture called the *bank of Wiener models* (BWMs), where multiple Wiener models are constructed in parallel as a *multiple-input multiple-output* (MIMO) system. With K models in the bank, we immediately see that this arrangement may increase the dimension of the projection space to K as long as the outputs are linearly independent. If we select this K-D space for nonlinear regression or adaptive filtering, we may get better fits. In fact, BWMs created by K Wiener models share the same structure as the first layer of a *time-delay neural network* (TDNN) [4] with K hidden units. Neural networks use *backpropagation* (BP) of the error for adaption. Without using BP, the existing method in system identification uses particle filtering and *expectation maximization* (EM) to develop an empirical optimization scheme [5, 6]. Alternatively, the BWMs structure can be trained by creating a mixture of experts [7] employing gating functions. In this paper, we propose a novel cost function based on *information-theoretic learning* (ITL) [8] that can be optimized by gradient ascent, which is also much faster and presents excellent accuracy.

Quadratic mutual information (QMI) have been broadly used in signal processing and machine learning applications [9, 10] to create empirical loss functions that utilize the full statistics to characterize the statistical dependency between the outputs and the desired responses. In this paper, we introduce a new implementation called the *empirical embedding of QMI* (E-QMI) that greatly reduces the computation time in optimizing QMI. By introducing two new types of normalization schemes, we show that E-QMI has the full potential to train BWMs. Experimentally we show that the BWMs trained by E-QMI is comparable to state-of-the-art models such as a TDNN. While TDNN uses backpropagation, our method has only feed-forward computations.

2. QUADRATIC MUTUAL INFORMATION

We first introduce the inspiration behind our new cost function and why it's used to train our model.

Given samples $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^p$. Suppose our samples are sampled from a fixed distribution $\mathbb{P}_\mathbf{x}$, the density of any $\mathbf{x}$ in the sample space can be estimated by $\hat{\pi}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^N K(\mathbf{x} - \mathbf{x}_n)$. A multivariate Gaussian function is frequently used as the function K. For any $\mathbf{x}_i, \mathbf{x}_j$ in the sample space, it is given by:

$$k_\sigma(\mathbf{x}_i - \mathbf{x}_j) = \frac{1}{(2\pi)^{p/2} \cdot \sigma^p} \exp(-\frac{1}{2\sigma^2} \cdot \|\mathbf{x}_i - \mathbf{x}_j\|_2^2). \quad (1)$$

The quadratic entropy estimator is derived based on this kernel

This work was partially supported by NSF award 1747783.

density function, written as:

$$V_E(\mathbf{X}) = -\log_2(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k_\sigma(x_i, x_j)), \tag{2}$$

which is a sample-based quantity that evaluates the 'flatness' of a given density function. It's been known this estimator is tied to Renyi's formula of quadratic entropy.

Given sample pairs $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=0}^{N}$, where $\mathbf{x}_n \in \mathbb{R}^p$ and $\mathbf{y}_n \in \mathbb{R}^q$, we define the quadratic joint entropy estimator between $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ and $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^{N}$ as:

$$V_J(\mathbf{X}, \mathbf{Y}) = -\log_2(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k_\sigma(x_i, x_j) \cdot k_\sigma(y_i, y_j)). \tag{3}$$

Recall the formula of Shannon's mutual information, an estimator that evaluates the 'dependency' between $\{\mathbf{x}_n\}_{n=1}^{N}$ and $\{\mathbf{y}_n\}_{n=1}^{N}$ can be defined as:

$$I_Q(\mathbf{X}, \mathbf{Y}) = V_E(\mathbf{X}) + V_E(\mathbf{Y}) - V_J(\mathbf{X}, \mathbf{Y}). \tag{4}$$

This expression above is one form of the *quadratic mutual information* (QMI) [11]. Another form by computing the divergence between the joint and the marginal distributions can be found in [9, 10].

QMI has been broadly applied in signal processing and machine learning [9, 10, 12]. However, to maximize QMI as an objective function, the computation of $I(\mathbf{X}, \mathbf{Y})$ and its gradient has $O(n^2)$ complexity. In order to ease the computational complexity, we introduce the following estimator that instead uses batches to estimate a biased value of QMI.

3. EMPIRICAL EMBEDDING OF QUADRATIC MUTUAL INFORMATION

Given a probability space and a fixed distribution $\mathbb{P}_\xi : \mathbb{R}^{d_\xi} \mapsto [0, 1]$ of a d_ξ-dimensional random vector, we draw two independent random vectors $\xi_1 \in \mathbb{R}^{d_\xi}$ and $\xi_2 \in \mathbb{R}^{d_\xi}$ from $\mathbb{P}_\xi$. Now we define the following quantity ν_E as a functional of $\mathbb{P}_\xi$:

$$\nu_E(\mathbb{P}_\xi) = -\log_2 \mathbb{E}_{\xi_1 \sim \mathbb{P}_\xi, \xi_2 \sim \mathbb{P}_\xi}[k_\sigma(\xi_1 - \xi_2)]. \tag{5}$$

Given a d_ξ-dimensional random vector and a d_η-dimensional random vector, let their joint distribution $\mathbb{P}_{\{\xi,\eta\}} : \mathbb{R}^{d_\xi} \times \mathbb{R}^{d_\eta} \mapsto [0, 1]$ be given. Let their marginal distributions be $\mathbb{P}_\xi$ and $\mathbb{P}_\eta$. We draw two pairs of random variables $\{\xi_1, \eta_1\}$ and $\{\xi_2, \eta_2\}$ from $\mathbb{P}_{\{\xi,\eta\}}$. We define the quantity ν_J that characterizes the joint distribution $\mathbb{P}_{\{\xi,\eta\}}$:

$$\begin{aligned} \nu_J(\mathbb{P}_{\{\xi,\eta\}}) &= -\log_2 \mathbb{E}[k_\sigma(\xi_1 - \xi_2) \cdot k_\sigma(\eta_1 - \eta_2)], \\ &\text{where } \{\xi_1, \eta_1\} \sim \mathbb{P}_{\{\xi,\eta\}}, \{\xi_2, \eta_2\} \sim \mathbb{P}_{\{\xi,\eta\}}. \end{aligned} \tag{6}$$

We now define the *empirical embedding of quadratic mutual information* (E-QMI) as follows:

$$I_{EQ}(\mathbb{P}_{\{\xi,\eta\}}) = \nu_E(\mathbb{P}_\xi) + \nu_E(\mathbb{P}_\eta) - \nu_J(\mathbb{P}_{\{\xi,\eta\}}). \tag{7}$$

While QMI characterizes the statistical dependency between two given realizations, E-QMI characterizes the dependency between two random vectors sampled from a certain joint distribution. In practice, we usually have full access to the realizations. However, we often compute the stochastic gradient with sample batches in optimization. Although we're not computing the exact value of QMI and its gradient, this new objective function will greatly accelerate the training process.

4. E-QMI FOR REGRESSION TASKS

We start from the one-dimensional case. Given a pair of random variables $\{\mathbf{x}, \mathbf{d}\}$, with $\mathbf{x} \in \mathbb{R}^L$ sampled from $\mathbb{P}_\mathbf{x}$ and $\mathbf{d} \in \mathbb{R}$ sampled from $\mathbb{P}_\mathbf{d}$. Let the joint distribution be $\mathbb{P}_{\{\mathbf{x},\mathbf{d}\}}$. The goal is to learn a function $f : \mathbb{R}^L \to \mathbb{R}$ such that $I_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}})$ is maximized. Observe that $I_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}})$ can be written as:

$$\begin{aligned} &I_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}}) = \\ &-\log_2\{\frac{\mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))] \cdot \mathbb{E}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)]}{\mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2)) \cdot k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)]}\}. \end{aligned} \tag{8}$$

By Cauchy-Schwarz inequality, we have $\mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))]\cdot\mathbb{E}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)] \leq \mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))\cdot k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)]$. Therefore $I_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}}) \geq 0$ holds for any measurable and feasible function f. However, simply maximizing I_{EQ} will be unstable, since the gain of f is not constrained and thus I_{EQ} is not bounded. One can constrain the norm of the weight vector as being introduced in [11] to keep the scale of $f(\mathbf{x})$ commensurate with $\mathbf{d}$. Here we introduce two new normalization schemes for this purpose.

4.1. Type-I normalization

We simply normalize $f(\mathbf{x})$ by its standard deviation to match the standard deviation of $\mathbf{d}$ to keep $f(\mathbf{x})$ and $\mathbf{d}$ in the same scale. Let the target standard deviation std$[\mathbf{d}]$ be given, we define the corresponding new optimum as:

$$I^*_{EQ} = \sup_f I_{EQ}(\mathbb{P}_{\{\frac{f(\mathbf{x})}{\text{std}[f(\mathbf{x})]}, \frac{\mathbf{d}}{\text{std}[\mathbf{d}]}\}}). \tag{9}$$

We call $I_{EQ}(\mathbb{P}_{\{\frac{f(\mathbf{x})}{\text{std}[f(\mathbf{x})]}, \frac{\mathbf{d}}{\text{std}[\mathbf{d}]}\}})$ the Type-I cost function.

4.2. Type-II normalization

Observe that if $f(\mathbf{x})$ and $\mathbf{d}$ are strictly dependent and $f(\mathbf{x}) = \mathbf{d}$, equation 8 can be written as $I_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}} = \mathbb{P}_{\{f(\mathbf{x}),f(\mathbf{x})\}}) = -\log_2\{\frac{\mathbb{E}[k_\sigma(f(\mathbf{x}_1)-f(\mathbf{x}_2))]^2}{\mathbb{E}[k_\sigma(f(\mathbf{x}_1)-f(\mathbf{x}_2))^2]}\}$.

If we optimize over f that satisfies $\frac{\mathbb{E}[k_\sigma(f(\mathbf{x}_1)-f(\mathbf{x}_2))]^2}{\mathbb{E}[k_\sigma(f(\mathbf{x}_1)-f(\mathbf{x}_2))^2]} = \frac{1}{2}$, then the optimum will be bounded by 1 from above. Now we want to impose this constraint by adding a constant. By solving the equation:

$$\begin{aligned} 2 \cdot \mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2)) + b]^2 \\ = \mathbb{E}[(k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2)) + b)^2], \end{aligned} \tag{10}$$

we obtain the solution $b = \text{std}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))] - \mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2)]$. For simplicity, we denote $\mathbf{z}_f = k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2)) - \mathbb{E}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))] + \text{std}[k_\sigma(f(\mathbf{x}_1) - f(\mathbf{x}_2))]$ and $\mathbf{z}_d = k_\sigma(\mathbf{d}_1 - \mathbf{d}_2) - \mathbb{E}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)] + \text{std}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)]$. We define the Type-II cost function as:

$$i_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}}) = -\log_2\left[\mathbb{E}[\mathbf{z}_f] \cdot \mathbb{E}[\mathbf{z}_d]\right] + \log_2\left[\mathbb{E}[\mathbf{z}_f \cdot \mathbf{z}_d]\right]. \tag{11}$$

Now we have $i^*_{EQ} = \sup_f i_{EQ}(\mathbb{P}_{\{f(\mathbf{x}),\mathbf{d}\}}) \leq 1$, which means the new optimum will be bounded from above by 1.

5. TRAINING A BANK OF WIENER MODELS THROUGH E-QMI FOR REGRESSION

Now we show that these two forms of E-QMI can be used to train the *bank of Wiener models* (BWMs).

Given $k \in \{1, 2...K\}$, an element of the bank $h_{\theta_k} : \mathbb{R}^L \to \mathbb{R}$ consists of a linear block $l_{\theta_k} : \mathbb{R}^L \to \mathbb{R}$ and a nonlinear block $\sigma : \mathbb{R} \to \mathbb{R}$. Here K stands for the number of models and L stands for the order of each model. Each element h_{θ_k} is exactly a Wiener model. The k-th Wiener model is parameterized by $\theta_k = (\mathbf{w}_k, b_k)$, where $\mathbf{w}_k \in \mathbb{R}^L$ and $b_k \in \mathbb{R}$. Given a random vector $\mathbf{x} \in \mathbb{R}^L$ as the input signal, the output of the k-th Wiener model is written as $h_{\theta_k}(\mathbf{x}) := \mathbf{z}_{\theta_k} = \sigma(\mathbf{w}_k^\intercal \mathbf{x} + b_k)$. The outputs of the BWMs form a vector $\mathbf{z}_{\theta_{1:K}} = [\mathbf{z}_{\theta_1}, \mathbf{z}_{\theta_2} ... \mathbf{z}_{\theta_K}]^\intercal$. Let the target signal $\mathbf{d} \in \mathbb{R}$ be given.

For regression or adaptive filtering, we first train the BWMs with two types of E-QMI w.r.t. the desired signal. After the model parameters are fixed, we find the best projection of $\mathbf{d}$ onto the linear space spanned by $\mathbf{z}_{\theta_{1:K}}$ by computing the *least square* (LS) solution.

5.1. Type-I normalization

Following the one-dimensional case, we normalize each $\mathbf{z}_{\theta_k}$ by its standard deviation. Let $\mathbf{z}'_{\theta_{1:K}} = [\frac{\mathbf{z}_{\theta_1}}{\text{std}[\mathbf{z}_{\theta_1}]}, \frac{\mathbf{z}_{\theta_2}}{\text{std}[\mathbf{z}_{\theta_2}]} \cdots \frac{\mathbf{z}_{\theta_K}}{\text{std}[\mathbf{z}_{\theta_K}]}]^\intercal$ and $\mathbf{d}' = \frac{\mathbf{d}}{\text{std}[\mathbf{d}]}$, we propose the following maximization problem:

$$\underset{\{\theta_1,\theta_2...\theta_K\}}{\text{maximize}} \quad I_{EQ}(\mathbb{P}_{\{\mathbf{z}'_{\theta_{1:K}},\mathbf{d}'\}}) \tag{12}$$

Let the batch size M be given, we sample two batches $\{x_m^{(1)}, d_m^{(1)}\}_{m=1}^M$ and $\{x_m^{(2)}, d_m^{(2)}\}_{m=1}^M$ from the dataset. Each term in E-QMI and its gradient is estimated empirically by the metric function between $\{x_m^{(1)}, d_m^{(1)}\}_{m=1}^M$ and $\{x_m^{(2)}, d_m^{(2)}\}_{m=1}^M$. We also estimate the standard deviation empirically by $\text{std}'[f(\mathbf{x})] = \sqrt{\frac{1}{2M}\sum_{m=1}^M (f(x_m^{(1)}) - f(x_m^{(2)}))^2}$. The parameter set $\{\theta_1, \theta_2 ... \theta_K\}$ is adapted by gradient ascent.

5.2. Type-II normalization

Followed by the one-dimensional case, we propose the following maximization problem:

$$\underset{\{\theta_1,\theta_2...\theta_K\}}{\text{maximize}} \quad i_{EQ}(\mathbb{P}_{\{\sum_{k=1}^K \mathbf{z}_{\theta_k},\mathbf{d}\}}). \tag{13}$$

The estimation of E-QMI and the standard deviation follows the same procedure as in Type-I.

5.3. Adaptive estimation

Estimating the standard deviation and its gradient by batches will introduce a bias at each time step. Inspired by neural network optimizer [13], we introduce an adaptive estimation scheme for Type-II that greatly reduces the variance in the training. Let $\mathbf{k}_z = k_\sigma(\sum_{k=1}^K \mathbf{z}_{\theta_k}^{(1)} - \sum_{k=1}^K \mathbf{z}_{\theta_k}^{(2)})$ and $\mathbf{k}'_d = k_\sigma(\mathbf{d}_1 - \mathbf{d}_2) - \mathbb{E}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)] + \text{std}[k_\sigma(\mathbf{d}_1 - \mathbf{d}_2)]$, we denote $m_z = \mathbb{E}[\mathbf{k}_z]$, $s_z = \text{std}[\mathbf{k}_z]$. Let $m_d = \mathbb{E}[\mathbf{k}'_d]$ be given. We also write $\mathbf{k}'_z = \mathbf{k}_z - m_z + s_z$ and $m_c = \mathbb{E}[\mathbf{k}'_z \cdot \mathbf{k}'_d]$. The partial derivative of i_{EQ} respect to θ_k is as follows:

$$\begin{aligned}\frac{\partial i_{EQ}}{\partial \theta_k} = &-\frac{1}{m_c}\mathbb{E}\big[[\mathbf{k}'_d + m_d \cdot (\frac{\mathbf{k}_z}{s_z} - \frac{m_z}{s_z} - 1)]\frac{\partial \mathbf{k}_z}{\partial \theta_k}\big] \\ &+ \frac{1}{s_z}\mathbb{E}[(\frac{\mathbf{k}_z}{s_z} - \frac{m_z}{s_z})\frac{\partial \mathbf{k}_z}{\partial \theta_k}].\end{aligned} \tag{14}$$

Given $t \in \{1, 2...\}$ for the t-th update of the gradient ascent. Let the batch estimation at time t be $\tilde{m}_z(t)$, $\tilde{s}_z(t)$ and $\tilde{m}_c(t)$. Let β_1, β_2 and β_3 be given, we create a sequence $\hat{m}_z(t)$, $\hat{s}_z(t)$ and $\hat{m}_c(t)$ such that $\hat{m}_z(t) = \beta_1 \hat{m}_z(t-1) + (1-\beta_1)\tilde{m}_z(t)$, $\hat{s}_z^2(t) = \beta_2 \hat{s}_z^2(t-1) + (1-\beta_2)\tilde{s}_z^2(t)$, and $\hat{m}_c(t) = \beta_3 \hat{m}_c(t-1) + (1-\beta_3)\tilde{m}_c(t)$, where $\hat{m}_z(0) = \hat{m}_c(0) = \hat{s}_z(0) = 0$. Then we use $\frac{\hat{m}_z(t)}{1-\beta_1^t}$, $\frac{\hat{s}_z(t)}{\sqrt{1-\beta_2^t}}$, and $\frac{\hat{m}_c(t)}{1-\beta_3^t}$ to estimate equation 14. We found that by tracking these three scalar-valued statistics, the learning curves are smooth and consistent.

6. RESULTS

6.1. Performance on regression tasks

Now we present the results of the following two tasks:

Frequency doubler (FD): Let $n \in \{1, 2...\}$ be given, we have the input signal $x_n = \sin(0.02 \cdot \pi n)$ and the target signal $d_n = \sin(0.04 \cdot \pi n)$. Ths mapping from x_n to d_n is nonlinear. The setting is simple, but it is also one important challenge that cannot be solved by a single Wiener model.

Lorenz system (LORENZ): Given the triple $\{x_t, y_t, z_t\}$ generated by Lorenz system with coefficients $\{\sigma = 10, \rho = 28, \beta = 2.667\}$ and initial states $\{x_0 = 0, y_0 = 1, z_0 = 1.05\}$. Let $\{x_n, y_n, z_n\}$ be the discrete signals sampled from $\{x_t, y_t, z_t\}$ with 100Hz. We treat x_n as the hidden state, signal y_n as the observation and z_n as the target. The signals produced by the system is highly nonstationary, and the mapping from y_n to z_n is also highly nonlinear.

Model parameters: Let the model order be L, and the number of models be K. We compare the BWMs trained by Type-I (T-I) and Type-II (T-II) with TDNN trained by backpropagation and MSE. TDNN has one single hidden layer with K units and an input dimension L. We fix $\{L = 3, K = 3\}$ for FD when using TDNN and T-II, and $\{L = 10, K = 3\}$

when using T-I for stabler results. We fix $\{L = 10, K = 3\}$ for LORENZ. We use adaptive estimation in section 5.3 for T2, and Adam optimizer [13] otherwise. We fix hyperparamters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\beta_3 = 0.999$. The learning rate is selected from $\{0.01, 0.1, 1\}$ to produce the best score. The result is shown below.

Table 1. The results of regression tasks

	LORENZ			FD		
	MSE	EQMI (T-I)	EQMI (T-II)	MSE ($\times 10^{-4}$)	EQMI (T-I)	EQMI (T-II)
TDNN	0.017	0.156	0.784	8.0	0.222	0.999
BWMs (T-I)	0.022	0.164	0.763	7.0	0.225	0.999
BWMs (T-II)	0.017	0.157	0.791	7.8	0.222	0.999

As can be seen, BWMs trained by E-QMI are very competitive to TDNN trained by BP. Our methodology avoids using BP and takes full advantage of the *pdf* information between the processed input signal and the desired response. MSE is unable to to train an MIMO system because it only finds the best linear projection onto the space, i.e, the Wiener solution. Also notice that the second linear projection layer created for T-II is close to equal weighting, while the LS solution is computed when using T-I and it's not equal weighting. It also shows that both E-QMI quantities can be used to evaluate the model performance. Another point to make is that the units of MSE is the error power, while the units of E-QMI are bits.

6.2. Illustrative comparisons

Here we present three illustrative comparisons to show the advantages of our method.

(1) Speed test: Given a dataset with one-dimensional samples, we compute the exact value of QMI with a subset with N samples. Then we sample two batches with N samples to compute E-QMI using equation 7. The figure below shows the speed comparison using an Intel i7-8550U CPU.

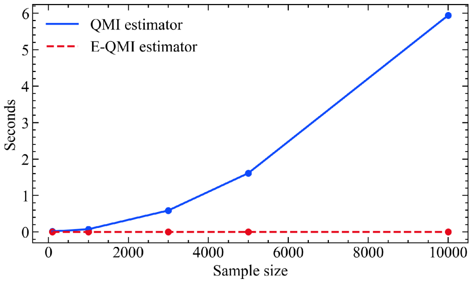
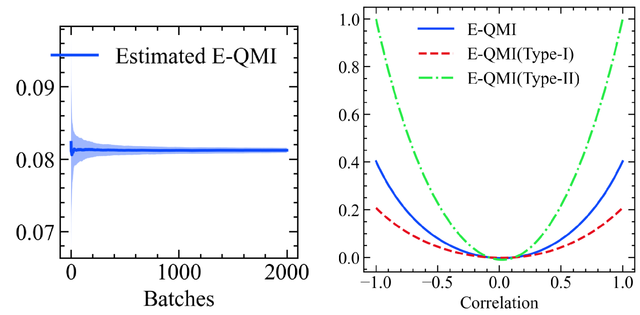


Fig. 1. Speed test of E-QMI.

(2) Effectiveness of E-QMI: We construct a sequence of Gaussian distribution pairs with their correlation ranging from -1 to 1. We first evaluate the variance of estimating E-QMI in this setting using multiple batches, where each batch contains 1000 samples. We also compare the value given by E-QMI with and without normalization to show that they all effectively characterize the dependency between two distributions.

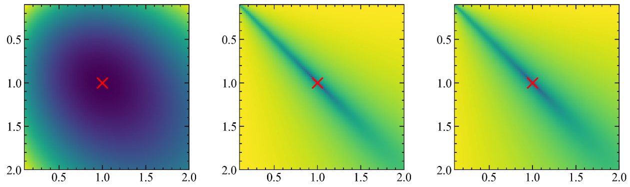


(a) The variance is small, and the total time for computing 2000 batches is around 0.3 second.

(b) All three quantities characterize the statistical dependency between distributions

Fig. 2. We show that E-QMI is both accurate and effective.

(3) Loss surfaces: We compare the loss surfaces given by MSE and E-QMI. For display, we creat a toy example where $x_1 = \sin(\pi n/500)$ and $x_2 = \cos(\pi n/500)$ for $n \in \{1, 2...\}$. We create a target signal $d = \text{sigmoid}(x_1 + x_2)$. Let the model be $y = \text{sigmoid}(w_1 \cdot x_1 + w_2 \cdot x_2)$, we compare the loss surface of this system identification problem respect to w_1 and w_2. For Type-I and Type-II, we take negative log value of E-QMI. The red cross in the figure shows the optimal parameters $w_1 = 1, w_2 = 2$. We found that both E-QMI quantities preserve the global optimum.



(a) MSE (b) Type-I ($\sigma = 0.4$) (c) Type-II

Fig. 3. Comprison of loss surfaces given by MSE and E-QMI. We also found that Type-II is more invariant to the kernel size σ while Type-I is more sensitive to σ.

7. CONCLUSION

This paper shows a way to train a *bank of Wiener models* (BWMs) with a cost function called the *empirical embedding of quadratic mutual information* (E-QMI) that utilizes the full statistics of the model outputs and the desired signal. We show that our approach provides equivalent performance to a single-hidden-layer TDNN trained with backpropagation. We found out that it is important to normalize the outputs of the BWMs for stable learning. Our empirical embedding methodology to estimate the E-QMI loss function is much faster and well behaved than the previous method of estimating QMI in information-theoretic learning literature. This methodology opens new avenues to adapt BWMs and may have an important impact in machine learning of sequential data streams.

8. REFERENCES

[1] Alex Simpkins. System identification: Theory for the user. *IEEE Robotics & Automation Magazine*, 19(2):95–96, 2012.

[2] Anna Hagenblad and Lennart Ljung. Maximum likelihood estimation of wiener models. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, volume 3, pages 2417–2418. IEEE, 2000.

[3] Anna Hagenblad. *Aspects of the identification of Wiener models*. Citeseer, 1999.

[4] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.

[5] Stuart Gibson and Brett Ninness. Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682, 2005.

[6] Adrian Wills, Thomas B. Schön, Lennart Ljung, and Brett Ninness. Identification of hammerstein–wiener models. *Automatica*, 49(1):70–81, 2013.

[7] Irwin W Sandberg, James T Lo, Craig L Fancourt, Jose C Principe, Shigeru Katagiri, and Simon Haykin. *Nonlinear dynamical systems: feedforward neural network perspectives*, volume 21. John Wiley & Sons, 2001.

[8] Jose C. Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[9] Shujian Yu, Matthew Emigh, Eder Santana, and Jose C. Principe. Autoencoders trained with relevant information: Blending shannon and wiener's perspectives. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6115–6119. IEEE, 2017.

[10] Dongxin Xu and Jose C. Principe. Training mlps layer-by-layer with the information potential. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 3, pages 1716–1720. IEEE, 1999.

[11] Luis G. Sanchez Giraldo and Jose C. Principe. Information theoretic learning with infinitely divisible kernels. *arXiv preprint arXiv:1301.3551*, 2013.

[12] Austin J. Brockmeier, John S. Choi, Evan G. Kriminger, Joseph T Francis, and Jose C. Principe. Neural decoding with kernel-based metric learning. *Neural Computation*, 26(6):1080–1107, 2014.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.