Heterogeneous model selection for multiscale time series forecasting

Radoslav Neychev
Moscow Institute of Physics and Technology, Yandex LLC
Moscow, Russia
nexes@yandex-team.ru

Anastasia Motrenko
Moscow Institute of Physics and Technology
Moscow, Russia
anastasia.motrenko@gmail.com

Eric Gaussier
Laboratory of Informatics in Grenoble
Grenoble, France
Eric.Gaussier@imag.fr

Vadim Strijov
Moscow Institute of Physics and Technology, CC RAS
Moscow, Russia
strijov@ccas.ru

ABSTRACT

A computational experiment to select an optimal model requires consequent trials of models. Due to highly complex dependencies in the large set of multiscale time series, various forecasting models are in the competitive set to choose from. The paper investigates a problem of models selection. It proposes a new method to compare heterogeneous models and to select an optimal subset. This subset acts in ensemble to boost forecasting quality. The models are experts in the mixture. Their likelihood is the essential criterion to compare. A gating function computes the likelihood. The experiment uses linear model, ensemble of decision trees, SVM and gradient boosting as the experts. Multiscale sets of industrial and weather time series illustrate this method with application to the IoT.

CCS CONCEPTS

•Applied computingft →Forecasting; •Computing methodologies →Ensemble methods;

KEYWORDS

model selection, time series forecasting, mixture of experts, gating function, heterogeneous model

ACM Reference format:
Radoslav Neychev, Anastasia Motrenko, Eric Gaussier, and Vadim Strijov. 2017. Heterogeneous model selection for multiscale time series forecasting. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, August, 2017 (SIGIR'17),* 4 pages.
DOI: 00.000/000_0

1 INTRODUCTION

To help IT support users to predict system state and failures in order to perform some preventive maintenance monitoring systems consider a continuous flow of data. It comes from numerous and various sensors of devices from the Internet of Things. It manages the large scale of measured data. These measurements are performed at various frequencies. Time series may have missing values [6] due to down time or due to a device misconfiguration. Time series analysis [9] and forecasting [13] comes out as the best candidate to model these big temporal monitoring data.

Building time series forecasts based for large dataset is complex because of different types of dependencies within feature space. Therefore using models in ensemble increases the forecasting quality. Constructing ensembles of models in various applications [1, 11] is a well-developed technique. The voting procedures, which combine, average or vote the model forecasts with some weights, are gradient boosting [3] and Random Forest [4]. The voting weights of these procedures are assigned to the models and do not depend on time or local variations in time series. Each weight keeps its value after optimization on a train dataset.

To make models take responsibility for a particular time or a time series variation, one uses a gating function. It brings the Mixture of Experts approach [12]. It decreases the computational complexity of the forecasting model without a quality loss [11]. Previously this function appeared in various forms: as simple softmax, as Dirichlet Process [8], neural-network [11], etc. The forecasting problem solutions use gating of the Mixture of Experts in [2, 10].

This paper treats the weights on the gating function as the likelihood of the ensemble models. It investigates the case of using several heterogeneous models as experts to answer the question: which experts are *the most useful* and *how many experts are optimal*? The model selection procedure includes the robust and most likely models into the ensemble. This paper proposed an approach to estimate the likelihood, computed by the gating function. The model selection procedure is based on the expert scopes in data and the likelihood variance. If the model fails to describe the dataset well it must be pruned.

The proposed method is tested on real-life data. The experiment is held on energy and weather data in Poland [5], which also has complex structure and cross-correlations [7], the other large sets of multiscale time series are to be used as well.

2 FORECASTING PROBLEM STATEMENT

There given a large set of time series $\mathfrak{D} = \{\mathbf{s}^q\}$, $\mathbf{s} \in \mathbb{R}^T$, $q = 1, \dots, Q$, where each time series

$$\mathbf{s} = [s_1, \dots, s_i, \dots, s_T]$$

SIGIR'17, Tokyo, Japan

DOI: 00.000/000_0

is a sequence of observations $s_i = s(t_i)$. Each time series $\mathbf{s}$ has its own sampling rate $1/\tau^{(q)}$: $t_i^{(q)} = i \cdot \tau^{(q)}$.

The problem is to obtain forecasts $\hat{s}(t_i)$ for $T_{\max} < t_i \leq T_{\max} + \Delta t_{\mathrm{r}}$. The forecasts $\hat{\mathbf{s}}$ should minimize the error function. This paper uses MAE and MAPE functions:

$$\mathrm{MAE} = \frac{1}{r}\|\hat{\mathbf{s}} - \mathbf{s}\|_1, \quad \mathrm{MAPE} = \frac{1}{r}\Big\|\frac{\hat{\mathbf{s}} - \mathbf{s}}{\mathbf{s}}\Big\|_1.$$

2.1 Design matrix construction

Represent the forecasting problem as the multiscale autoregression problem, where target variables are the vectors of lagged values $s(t_i)$.

Denote by $\mathbf{z}$ a row of the design matrix $\mathbf{Z}$. It collects the time series $\mathbf{s}^q$ over some time period Δt_{p}. The vector $\mathbf{z}$ includes samples from the history of time series from $\mathfrak{D}$.

The design matrix $\mathbf{Z}$ for the multiscale autoregressive problem statement is constructed as follows. Let denote $\mathbf{s}_i^q$ by i-th segment of the time series $\mathbf{s}^q$

$$[\mathbf{x}_i^q | \mathbf{y}_i^q] = \tag{1}$$

$$\underbrace{s^q(t_i - \Delta t_{\mathrm{r}} - \Delta t_{\mathrm{p}}), \dots,}_{\mathbf{x}_i^q} \underbrace{s^q(t_i - \Delta t_{\mathrm{r}}), \dots, s^q(t_i)}_{\mathbf{y}_i^q}],$$

where $s^q(t)$ is an element of time series $\mathbf{s}^q$. To construct the design matrix, select t_i, $i = 1, \dots, m$ from $G = \{t_1, \dots, t_T\}$ such that segments $\mathbf{s}_i = [\mathbf{x}_i | \mathbf{y}_i]$ cover time series $\mathbf{s}$ without intersection in target parts $\mathbf{y}_i$:

$$|t_{i+1} - t_i| > \Delta t_{\mathrm{r}}. \tag{2}$$

Following (1) and (2), split segments $[\mathbf{x}_i^{(q)} | \mathbf{y}_i^{(q)}]$, $i = 1, \dots, m$ from all time series $\{\mathbf{s}^q\}$ and form the matrix

$$\mathbf{Z} = \left[\begin{array}{c|c} \underset{1\times n}{\mathbf{x}} & \underset{1\times r}{\mathbf{y}} \\ \hline \underset{m\times n}{\mathbf{X}} & \underset{m\times r}{\mathbf{Y}} \end{array}\right] \tag{3}$$

Denote a row from the pair $\mathbf{X}, \mathbf{Y}$ as $\mathbf{x}, \mathbf{y}$ and call these vectors the features and the target. State the regression problem:

$$\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{w}}), \qquad \hat{\mathbf{w}} = \underset{\hat{\mathbf{w}}}{\arg\min}\, S\big(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}\big), \tag{4}$$

where $\mathbf{f}$ is a forecasting model with parameters $\mathbf{w}$ and S is the error function MSE.

2.2 Rolling validation procedure

To test forecast model on flow data use the *rolling validation procedure* (5). The procedure makes K forecasts $\mathbf{y}_k$ for segments $\mathbf{x}_k$. Each segment $\mathbf{x}_k$ starts at time-point t_k. It has a fixed length Δt_k. To construct a design matrix for each time-point t_k split time-scale into K segments Δt_{r} up to the end. For each time-point t_k do the following:

1) construct the validation vector $\mathbf{x}^*_{\mathrm{val},k}$ for time series of the length Δt_{r} as the first row of the design matrix $\mathbf{Z}$,
2) construct the rest rows of the design matrix $\mathbf{Z}$ for the time after t_k and present it as

$$\mathbf{Z} = \left[\begin{array}{c|c} \dots & \dots \\ \hline \underset{1\times n}{\mathbf{x}_{\mathrm{val},k}} & \underset{1\times r}{\mathbf{y}_{\mathrm{val},k}} \\ \underset{m_{\min}\times n}{\mathbf{X}_{\mathrm{train},k}} & \underset{m_{\min}\times r}{\mathbf{Y}_{\mathrm{train},k}} \\ \hline \dots & \dots \end{array}\right], \Big\uparrow_k \tag{5}$$

3) optimize model parameters $\mathbf{w}, \beta$ using $\mathbf{X}_{\mathrm{train},k}, \mathbf{Y}_{\mathrm{train},k}$, (run additional cross-validation procedure on there rows to select the optimal model structure, if necessary),
4) optimize model parameters $\mathbf{w}$ using $\mathbf{X}_{\mathrm{train},k}, \mathbf{Y}_{\mathrm{train},k}$ and compute residues $\boldsymbol{\varepsilon}_k = \mathbf{y}_{\mathrm{val},k} - \mathbf{f}(\mathbf{x}_{\mathrm{val}_k}, \mathbf{w})$ and MAPE,
5) increase k and repeat.

3 MIXTURE OF EXPERTS

Assume the model $\mathbf{f}$ with gaussian noise $\boldsymbol{\varepsilon}$

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{w}) + \boldsymbol{\varepsilon}, \ \mathbf{f}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}, \ \mathbf{y} \sim \mathcal{N}(\mathbf{w}^{\mathsf{T}}\mathbf{x}, \beta).$$

Suppose that each model $f(\mathbf{x}, \mathbf{w}_k)$ generates a sample $(\mathbf{x}, \mathbf{y})$ with probability $p(k|\mathbf{x}, \mathbf{w})$. Then the following factorization holds

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(\mathbf{y}, k|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(k|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{y}|k, \mathbf{x}, \boldsymbol{\theta}) =$$

$$= p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}, \mathbf{v}_k) \mathcal{N}(\mathbf{y}|\mathbf{w}_k^{\mathsf{T}}\mathbf{x}, \beta), \tag{6}$$

where

$$\pi_k(\mathbf{x}, \mathbf{v}_k) = \frac{\exp(\mathbf{v}_k^{\mathsf{T}}\mathbf{x})}{\sum_{k'=1}^{K} \exp(\mathbf{v}_{k'}^{\mathsf{T}}\mathbf{x})}.$$

In general case, the model likelihood $\pi_k(\mathbf{x})$ can be more complex. Call the *gating function* a mapping $\boldsymbol{\pi} : \mathbf{X} \to [0; 1]^K$. It shows how model $\mathbf{f}_k$ is confident on a sample $\mathbf{x}_i$. Denote by $\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_K, \mathbf{V}, \boldsymbol{\beta}]$ the vector of hyperparameters, where $\mathbf{V}$ is the vector of the gating function parameters. With the likelihood of $\mathbf{f}_k$ model on input $(\mathbf{x}, \mathbf{y})$ denoted as $p(k|\mathbf{x}, \mathbf{w})$, the $\mathbf{y}$ distribution is

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(\mathbf{y}, k|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(k|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{y}|k, \mathbf{x}, \boldsymbol{\theta}) =$$

$$= \sum_{k=1}^{K} \frac{\exp(\mathbf{v}_k^{\mathsf{T}}\mathbf{x})}{\sum_{k'=1}^{K} \exp(\mathbf{v}_{k'}^{\mathsf{T}}\mathbf{x})} \exp\left(-\frac{1}{2\beta_k}\big(\mathbf{y} - \mathbf{f}_k(\mathbf{x}, \mathbf{w}_k)\big)^2\right).$$

Let γ_{ik} be the likelihood of $\mathbf{f}_k$ on input $\mathbf{x}_i$, matrix $\boldsymbol{\Gamma} = [\gamma_{ik}]$ shows model-sample likelihoods. Columns of matrix $\boldsymbol{\Gamma}$ are values of gating function $\boldsymbol{\pi}$ on dataset samples.

3.1 EM algorithm

To optimize the vector of hyperparameters $\boldsymbol{\theta}$ two-step iterative procedure can be used.

E-step: Using current estimations $\mathbf{w}_1^r, \dots, \mathbf{w}_K^r, \mathbf{V}^r, \boldsymbol{\beta}^r$ recompute matrix

$$\boldsymbol{\Gamma}^{(r+1)} = [\pi_1(\mathbf{X}), \dots, \pi_K(\mathbf{X})]$$

as following:

$$\gamma_{ik}^{(r+1)} = \mathsf{E}(z_{ik}) = p(k|\mathbf{x}_i, \boldsymbol{\theta}^{(r)}) = \tag{7}$$

$$= \frac{\pi_k(\mathbf{x}_i)\mathcal{N}(y_i|f_k(\mathbf{x}_i,\mathbf{w}_k^{(r)}),\beta^{(r)})}{\sum_{k'=1}^{K} \pi_{k'}(\mathbf{x}_i)\mathcal{N}(y_i|f(\mathbf{x}_i,\mathbf{w}_{k'}^{(r)}),\beta^{(r)}).}$$

M-step: Using new values of γ_{ik} models parameters can be re-estimated:

$$\mathbf{v}_k = \arg\max_{\mathbf{v}} \sum_{i=1}^{m} \gamma_{ik}^{r+1} \ln \pi_k(\mathbf{x}_i, \mathbf{v}),$$

$$\mathbf{w}_k = \arg\max_{\mathbf{w}_k} \left[-\sum_{i=1}^{m} \gamma_{ik}^{r+1} (\mathbf{y}_i - \mathbf{f}_k(\mathbf{x}_i, \mathbf{w}_k))^2 \right],$$

$$\beta_k = \arg\max_{\beta} \left[n \ln \beta - \sum_{i=1}^{m} \frac{1}{\beta} (\mathbf{y}_i - \mathbf{f}_k(\mathbf{x}_i, \mathbf{w}_k))^2 \right].$$

To initialize the experts they can be trained on data subsets The subsets can be chosen according to prior knowledge or randomly picked from training data.

4 MODEL SELECTION USING GATING FUNCTION

Call model *insignificant*, if it has insignificant likelihood on almost all samples from the training set.

Figure 1 illustrates the model selection procedure. The upper plot shows a dataset for univariate regression. It contains piecewise linear functions of four segments with added noise. The parameters of five linear models are optimized on this dataset. The lower plot shows the likelihood (8) of each expert over the dataset. The model number five has near-zero likelihood. It does not affect the quality of approximation. So the model is insignificant and should be pruned from the ensemble.

As the gating function returns the likelihood of every expert on data samples, a neural network may be used for this purpose. In this paper the gating function is a 3-layers NN of following structure: $\mathbf{f} = \mathbf{a}(\mathbf{h}_N(\ldots \mathbf{h}_1(\mathbf{x})))(\mathbf{w})$, where $\mathbf{h}_k$ are autoencoders and $\mathbf{a}$ is a softmax classifier:

$$\mathbf{f}(\mathbf{w}, \mathbf{x}) = \frac{\exp(\mathbf{a}(\mathbf{x}))}{\sum_j \exp(a_j(\mathbf{x}))}, \qquad \mathbf{a}(\mathbf{x}) = \mathbf{W}_2^{\mathsf{T}} \mathbf{tanh}(\mathbf{W}_1^{\mathsf{T}} \mathbf{x}), \qquad (8)$$

$$\mathbf{h}_k(\mathbf{x}) = \sigma(\mathbf{W}_k \mathbf{x} + \mathbf{b}_k),$$

where $\mathbf{w}$ minimizes the error function.

The model selection procedure:

1) initialize the experts on preselected or random samples from the training set,
2) tune the MoE ensemble with the EM algorithm (Sec. 3.1),
3) prune insignificant models and train the new ensemble containing only significant ones.

5 COMPUTATIONAL EXPERIMENT

A real-life dataset [5] is used to test the proposed method. It contains energy consumption and weather conditions in Poland through 2000-2004 (Fig. 2). The provided time series are:

- Energy consumption (per hour)
- Maximum temperature (per day)
- Minimum temperature (per day)
- Wind power(per day)
- Relative humidity (per day)
- Solar conditions (per day)

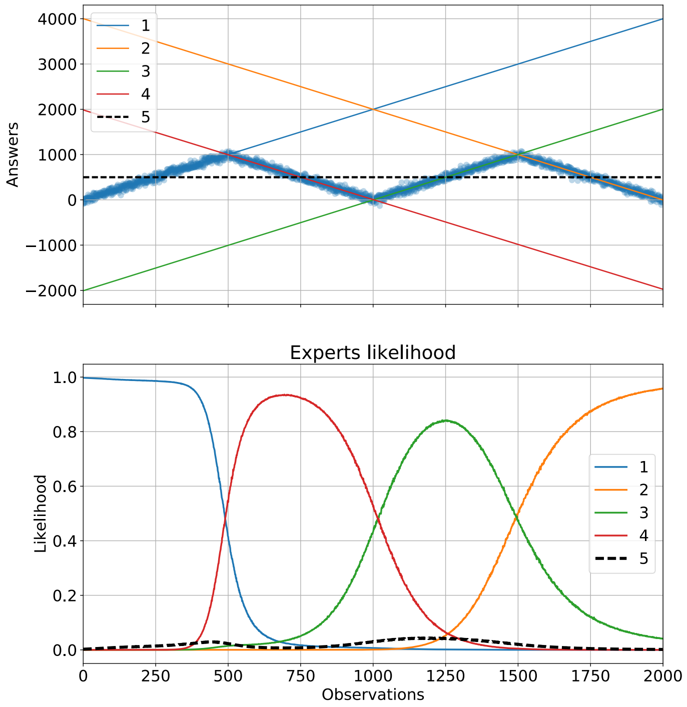

Figure 1: Five linear experts fitting the toy data

The target time series is the energy consumption. The goal is to predict energy consumption for the next 24 hours based on the last 6 days history. To solve the forecasting problem the next models were used:

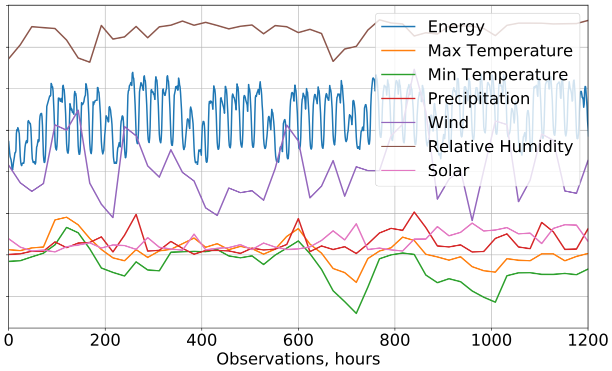

Figure 2: Time series visualization.

- Linear model (simple linear regression with no regularizers)
- Random forest regressor (with 50 estimators)
- Multiple Output XGBoost regressor
- Multiple Output SVR

The problem was solved by every model by itself and by the ensemble of the models. Table 1 shows the results.

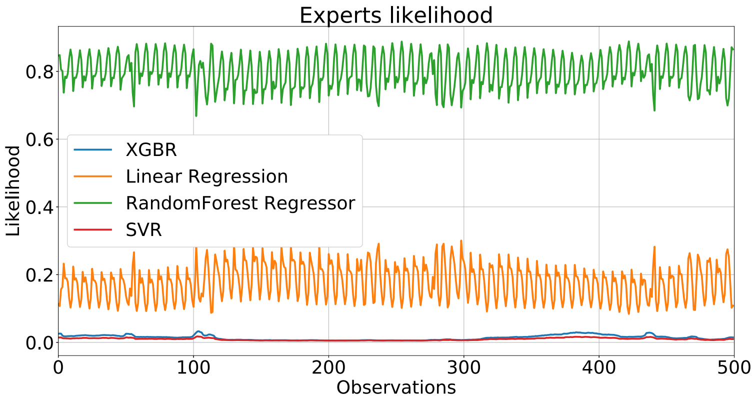

Figure 3: Likelihood of experts on energy-weather data.

Figure 3 shows the estimated likelihoods of the experts on the first 500 samples of the training set. The gating function (8) returns the likelihood of each expert. Each sample of the dataset is an observation. The samples come in chronological order and join samples of time series in the dataset [5]. Two models (SVR and XGBR) has near-zero likelihoods, therefore they are pruned out of the selected ensemble. The Linear and the Random forest models show significant likelihoods, which have one-week period. The Linear regression is more confident on weekdays (likelihood ≈ 0.3) than on weekends (likelihood ≈ 0.15). Random forest in contrast has higher likelihood (≈ 0.85) on weekends. These models complement each other. Their ensemble shows better quality and less (up to 10 times) computational time.

Table 1: Errors of models on train and test

Error	MAE		MAPE		Likelihood
Model	train	test	train	test	in ensemble
MO SVR	70014	68822	0.198	0.204	0.005
Linear reg	10738	16350	0.033	0.053	0.17
Random Forest	5501	17160	0.017	0.055	0.82
MO XGB	3563	17375	0.012	0.055	0.005
MoE (all)	6250	16641	0.020	0.054	–
MoE (RF + Lin Reg)	5656	**16120**	0.018	**0.052**	–

6 CONCLUSIONS

This paper proposes a method to select heterogeneous forecasting models using the gating function in the Mixture of Experts. An optimal ensemble of models makes forecast of large set of multiscale time series. Set of simple models appears to be more optimal, than one overcomplicated. The gating function defines the model likelihood on data samples. The computational experiment analyses the model selection results based on the values of the gating function. The proposed method prunes experts with insignificant likelihood in the ensemble. This method reduces computational complexity of the selected models and improve quality of forecasts. The experiment tested four different models: Support Vector Regression, Gradient Boosting (XGBR), Linear Regression and Random Forest. The achieved result shows that two models are insignificant in the ensemble. It shows better quality of forecasts and up to 10 times higher computational efficiency.

REFERENCES

[1] Quoc Le Barret Zoph. 2017. Neural Architecture Search with Reinforcement Learning. *ICLR* (2017).

[2] Faicel Chamroukhi. 2016. Skew-normal Mixture of Experts. *Neural Networks (IJCNN)* (2016).

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[4] Xi Chen and Hemant Ishwaran. 2012. Random Forests for Genomic Data Analysis. *Genomics* 99, 6 (2012), 323–329.

[5] Gregor Dudec. Electricity consumption and weather conditions dataset. (????).

[6] David S. Fung. 2006. Methods for the Estimation of Missing Values in TIme Series. *Edith Cowan University Thesis* (2006).

[7] A. Katrutsa and V. Strijov. 2015. Stress test procedure for feature selection algorithms. *Chemometrics* 142 (2015), 172–183.

[8] Carl Edward Rasmussen and Zoubin Ghahramani. 2002. Infinite Mixtures of Gaussian Process Experts. *Advances in Neural Information Processing Systems 14* (2002), 881–888.

[9] K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurtz. 2011. Comparison of correlation analysis techiques for irregularly sampled time series. *Nonlinear Processes in Geophysics* 18 (2011), 389–404.

[10] Rodrigo Arnaldo Scarpel. 2015. An integrated mixture of local experts model for demand forecasting. *International Journal of Production Economics* 164, C (2015), 35–42.

[11] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. (2017).

[12] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. 2012. Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems* 23, 8 (2012), 1177–1193.

[13] F. P ffierez Cruz, G. Camps-Valls, E. Soria-Olivas, J. P ffierez Ruixo, A. Figueiras-Vidal, and A. Art ffies Rodr ffiiguez. 2002. Multi-dimensional function approximation and regression estimation. *Artificial Neural Networks* (2002), 757–762.