A New Look at an Old Problem: A Universal Learning Approach to Linear Regression

Koby Bibas
School of Electrical Engineering
Tel Aviv University
Email: kobybibas@gmail.com

Yaniv Fogel
School of Electrical Engineering
Tel Aviv University
Email: Yaniv.fogel8@gmail.com

Meir Feder
School of Electrical Engineering
Tel Aviv University
Email: meir@eng.tau.ac.il

***Abstract*—Linear regression is a classical paradigm in statistics. A new look at it is provided via the lens of universal learning. In applying universal learning to linear regression the hypotheses class represents the label $y \in \mathcal{R}$ as a linear combination of the feature vector $x^T\theta$ where $x \in \mathcal{R}^M$, within a Gaussian error. The Predictive Normalized Maximum Likelihood (pNML) solution for universal learning of individual data can be expressed analytically in this case, as well as its associated learnability measure. Interestingly, the situation where the number of parameters M may even be larger than the number of training samples N can be examined. As expected, in this case learnability cannot be attained in every situation; nevertheless, if the test vector resides mostly in a subspace spanned by the eigenvectors associated with the large eigenvalues of the empirical correlation matrix of the training data, linear regression can generalize despite the fact that it uses an "over-parametrized" model. We demonstrate the results with a simulation of fitting a polynomial to data with a possibly large polynomial degree.**

I. Introduction

Linear regression, using least squares, is probably one of the most standard techniques in statistics, [1]. This work provides a new view of this problem based on recent results in universal learning. In particular, the common assumption in linear regression is that the number of training samples needs to be much higher than the number of features in order to be able to generalize [2]. Recently, the success of Deep Neural Networks (DNNs) in which the number of learnable parameters is in several orders of magnitudes greater than the size of the feature space requires rethinking that assumption. The new view we provide will show that sometimes generalization can be attained even in the "over-parameterized" regime.

Before diving into this analysis, a short introduction to universal learning is provided. In the common situation of supervised machine learning, a training set is given consisting of N pairs $z^N = \{(x_i, y_i)\}_{i=1}^N$, where $x \in \mathcal{X}$ is the data or the features and $y \in \mathcal{Y}$ is the label. Then, a new x is given and the task is to predict its corresponding label y. In the information theoretic framework considered in a variety of works, e.g., [3] and more recently [4], prediction is done by assigning a probability distribution $q(\cdot|x)$ to the unknown label, and the prediction loss is the log-loss:

$$\mathcal{L}(q; x, y) = -\log q(y|x). \tag{1}$$

Clearly a reasonable goal is to find the predictor q with the minimal loss for the test sample. However, this problem is ill-posed unless additional assumptions are made.

First, a "model" class, or 'hypotheses" class must be defined. This class is a set of conditional probability distributions

$$P_\Theta = \{p_\theta(y|x), \;\; \theta \in \Theta\} \tag{2}$$

where Θ is a general index set. This is equivalent to saying that there is a set of stochastic functions $\{y = g_\theta(x), \;\; \theta \in \Theta\}$ used to explain the relation between x and y.

Next, assumptions must be made on how the data and the labels are generated. In the stochastic setting, it is assumed that there is a true probabilistic relation between x and y given by an (unknown) model from the class P_Θ. A more general setting, used in the variety of works in machine learning, is the Probably Approximately Correct (PAC) established in [5]. In PAC x and y are assumed to be generated by some source $P(x, y) = P(x)P(y|x)$, but unlike the standard stochastic setting $P(y|x)$ is not necessarily a member of the hypothesis class. In both the stochastic and PAC settings the goal is to perform as well as a learner that knows the true probability.

The most general setting, however, and the one used in this paper is the individual, where the data and labels of both the training and test are specific and individual. In this setting the goal can no longer be to perform as well as a learner that knows the true probability. Instead, following [3], the goal is to seek a learner that can compete with a "genie" or a reference learner that knows the desired label value, but is restricted to use a model from the given hypotheses class P_Θ. In addition, as discussed in [4], the reference has no knowledge which of the samples is the test. Thus, the reference chooses

$$\hat{\theta}(z^N, x, y) = \arg\max_\theta \left[p_\theta(y|x) \cdot \Pi_{i=1}^N p_\theta(y_i|x_i)\right] \tag{3}$$

The log-loss difference between a universal learner q and the reference is the regret:

$$R(z^N, x, y, q) = \log \frac{p_{\hat{\theta}(z^N, x, y)}(y|x)}{q_(y|x; z^N)}. \tag{4}$$

As advocated in [4], the chosen universal learner solves:

$$\min_q \max_y R(z^N, x, y, q) = R^*(z^N, x) \tag{5}$$

Following [6] this learner, called the Predictive Normalized Maximum Likelihood (pNML), is given by

$$q_{\text{pNML}}(y|x;z^N) = \frac{p_{\hat{\theta}(z^n,x,y)}(y|x)}{\sum_{y\in\mathcal{Y}} p_{\hat{\theta}(z^n,x,y)}(y|x)}. \tag{6}$$

Note that this pNML probability assignment was essentially proposed earlier, see [7], [8], with a different motivation as one of the possible variations of the Normalized Maximum Likelihood (NML) method of [6] for universal prediction.

In order to obtain the pNML the following procedure is executed: assuming the label of the test data is known, find the best model that fits it with the training samples, and predict the assumed label by it. Repeat the process for all possible labels. Then, normalize to get a valid probability distribution which is the pNML learner. The regret of the pNML, $R^*(z^N,x)$ is the logarithm of its normalization factor

$$R^*(z^N,x) = \log\left\{\sum_{y\in\mathcal{Y}} p_{\hat{\theta}(z^n,x,y)}(y|x)\right\}. \tag{7}$$

In considering linear regression, $y \in \mathcal{R}$ is the scalar label, $x \in \mathcal{R}^M$ is the feature vector (sometimes the first component of x is set to 1 to formulate affine linear relation), and the model class is the set:

$$\left\{p_\theta(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right\},\ \ \theta\in\mathcal{R}^M\right\} \tag{8}$$

That is, the label y is a linear combination of the components of x, within a Gaussian noise. As shown below, in this case the pNML and its learnability measure can be evaluated explicitly.

The pNML approach deviates from the standard Empirical Risk Minimization (ERM) [9] approach. In ERM, given a training set and hypothesis class $\{p_\theta(y|x),\ \theta\in\Theta\}$, a learner that minimizes the loss over the training set is chosen:

$$q_{\text{ERM}}(y|x) = \underset{p_\theta}{\textit{argmin}}\ \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(p_\theta;x_i,y_i). \tag{9}$$

In the linear regression model (8), one chooses the least squares solution over the training set for the linear coefficients. This, however, may lead to large generalization log-loss error.

The paper has two main contributions. First, it provides an explicit analytical solution for the pNML learner and its "learnability" measure (which is the minmax regret (7)) for the linear regression hypothesis class. This includes also the regularized case where the norm of the coefficients vector is constrained. Second, based on the analysis of the learnability measure, it is shown that even in the over-parameterized case where the number of parameters M may be larger than the training size N, if the test data comes from a "learnable space" successful generalization occurs. This phenomenon may explain why other over-parameterized models such as deep neural networks are successful for "learnable" data.

The paper outline is as follows. Section II presents some related work. Section III provides the formal problem definition, while the pNML evaluation for the regression problem is given in sections IV and V. In-depth analysis of the learnable space is given in section VI. Simulation of the pNML and its regret for the problem of fitting a polynomial to data is described in section VII and the conclusions are in section VIII.

II. Related Works

In this section, we briefly mention related works on model generalization, least squares regression and the confidence of the least squares predictions.

Model Generalization. Understating the model generalization capabilities is considered a fundamental problem in machine learning [10]. As noted, most of the theoretical work in learning use the PAC setting. In that setting, a common measure is the VC Dimension that can be used to upper bound on the test generalization error. For DNNs, the VC dimension is linear with the number of parameters [11], yet the empirical evidence demonstrates that DNN's have state of the art generalization performance. This makes the VC dimension irrelevant for assessing the generalization error of DNNs.

Least Squares. The least squares algorithm is widely used in linear regression due to its robust performance and simplicity of implementation. In addition to the explicit formula for its solution, it can be solved sequentially, via the Recursive Least Squares (RLS) algorithm, which is an efficient online method for finding the linear predictor that minimizes the squared error over the training data [12]. In learning field, the least squares estimator is used for prediction in both ERM and Bayesian approaches [13]. This paper provides an analysis in the individual setting, in the realm of the pNML approach.

Outliers detection and confidence. In order to evaluate a pointwise confidence measure for linear regression, several methods were proposed. Leverage values are employed to identify outliers with respect to their feature values [14]. A leverage value is a measure of the distance between an observation and the center of the data

$$h_{ii} = x_i^T(XX^T)x_i \tag{10}$$

where X is the correlation matrix of the training set and x_i is the feature value which is examined. If the leverage value h_{ii} of observation is large, the observation is considered as an outlier. Using the pNML, we fit a learner assuming the label of the test sample is known. Therefore, testing wheater it is an outlier can produce prediction confidence. In section IV, we get analyticly similar expression to the leverage measure.

Another approach for finding the reliability of the prediction is to compose confidence intervals [15]. Confidence intervals are a pointwise measure that is sensitive to the variability of the features and sample size. Denote $\hat{y}$ as the predicted value of x and $\hat{\sigma}^2$ as the empirical error of the prediction, under the assumption of sthocastic i.i.d data and existence of white noise, a confidence interval convergences is distribution to

$$(\hat{y} - y) \to \mathcal{N}(0, \hat{\sigma}^2 x^T(XX^T)^{-1}x) \tag{11}$$

III. Linear Regression: Formal Problem Definition

Given N pairs of data and labels $\{x_i, y_i\}_{i=1}^N$ where $x_i \in R^M, y \in R$ are deterministic, the model takes the form:

$$\begin{aligned} y_1 &= x_1^T\theta + e_1 \\ &\vdots \\ y_N &= x_N^T\theta + e_N \end{aligned} \tag{12}$$

where $\theta \in R^M$ are the learnable parameters and the $e_i \in R$ are zero mean, Gaussian, independent with variance of σ^2. The goal is to predict y based on a new data sample x. Under the assumptions y, conditioned on x, has a normal distribution that depends on the learnable parameters θ

$$p_\theta(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y - x^T\theta\right)^2\right\}. \tag{13}$$

The unknown parameter vector θ belongs to a set Θ, which in the general case is the entire R^M. In the regularized version (that leads to Ridge regression [16]), Θ is the sphere $|\theta| \leq A$. In the next section, the pNML will be evaluated for this hypotheses class. Recall that the pNML learner of y given the the test sample x and the training set $z^N = \{(x_i, y_i)\}_{i=1}^N$ is given by:

$$q_{\text{pNML}}(y|x; z^N) = \frac{1}{\kappa} p_{\hat{\theta}(z^N;x,y)}(y|x). \tag{14}$$

where in the linear regression case

$$\hat{\theta}(z^N, x, y) = \arg\min_{\theta\in\Theta}\left[\sum_{i=1}^N \left(y_i - x_i^T\theta\right)^2 + \left(y - x^T\theta\right)^2\right] \tag{15}$$

and where κ is the the normalization factor:

$$\kappa = R^*(z^N, x) = \int_R p_{\hat{\theta}(z^N;x,y)}(y|x)dy, \tag{16}$$

The goal is to find an analytic expression for (14) and for the learnability measure $\Gamma = \log\kappa$, the minmax regret value.

IV. pNML Evaluation

The following notation is used. $X \in R^{M\times N+1}$ is the matrix which contains all the training data along with the test sample and $Y \in R^{N+1}$ is the vector which contains all the labels including the test label, i.e.,

$$X = \begin{bmatrix} x_1 & \dots & x_N & x \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \\ y \end{bmatrix} \tag{17}$$

Assuming that the test label y is given, the optimal solution under least squares:

$$\hat{\theta}(z^N, x, y) = \theta^*_{N+1} = (X^TX)^{-1}X^TY \tag{18}$$

By the Recursive Least Squares (RLS) formulation [12]:

$$\theta^*_{N+1} = \theta^*_N + P_{N+1}x(y - \hat{y}) \tag{19}$$

where $\hat{y} = x^T\theta^*_N$ is the ERM prediction based on the samples $\{(x_i, y_i)\}_{i=1}^N$ and

$$P_{N+1} = (XX^T)^{-1}. \tag{20}$$

Note that in RLS, P_{N+1} is also calculated recursively from P_N, but this is not needed at this point. Now,

$$\begin{aligned} P_{\theta^*_{N+1}}(y) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y - x^T\theta^*_{N+1}\right)^2\right\} = \\ &\frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big\{ - \frac{1}{2\sigma^2}\big(y - x^T\big(\theta^*_N + \\ &\qquad P_{N+1}x(y - \hat{y})\big)\big)^2\Big\} = \\ &\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(1 - x^TP_{N+1}x)^2}{2\sigma^2}(y - \hat{y})^2\right\}. \end{aligned} \tag{21}$$

To get the pNML normalization factor (16), we integrate over all possible labels

$$\begin{aligned} \kappa &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}}\ exp\left\{-\frac{(1 - x^TP_{N+1}x)^2}{2\sigma^2}(y - \hat{y})^2\right\} dy \\ &= \frac{1}{1 - x^TP_{N+1}x} = \frac{1}{1 - x^T(XX^T)^{-1}x} \end{aligned} \tag{22}$$

Thus, the pNML distribution of y given x is:

$$\begin{aligned} &q_{\text{pNML}}(y|x; z^N) = \frac{1}{\kappa} p_{\theta^*_{N+1}}(y|x) = \\ &\frac{1 - x^TP_{N+1}x}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(1 - x^TP_{N+1}x)^2}{2\sigma^2}(y - \hat{y})^2\right\} \end{aligned} \tag{23}$$

and its associate learnability measure or regret:

$$\Gamma = \log\kappa = \log\left(\frac{1}{1 - x^T(XX^T)^{-1}x}\right). \tag{24}$$

V. pNML with Regularization

Assuming now that the model class Θ is constrained to the sphere $|\theta| \leq A$, for some A. Using a Lagrange multiplier λ we get the Tikhonov regularization (or Ridge regression), where the expression to minimize is now:

$$\mathcal{L}(z^N) = \sum_{i=1}^N |y_i - x_i^T\theta|^2 + \lambda|\theta|^2 \tag{25}$$

With the test data, the "regularized" least square solution is:

$$\hat{\theta}(z^N, x, y) = \theta^*_{N+1} = (X^TX + \lambda I)^{-1}X^TY \tag{26}$$

Here too the RLS formula holds:

$$\theta^*_{N+1} = \theta^*_N + P_{N+1}x(y - \hat{y}) \tag{27}$$

However, now

$$P_{N+1} = (X^TX + \lambda I)^{-1}. \tag{28}$$

The rest of the evaluation can follow as before. Thus, the pNML learner in this "regularized" case is

$$q_{\text{pNML}}(y|x;z^N,\lambda) = \frac{1 - x^T(XX^T+\lambda I)^{-1}x}{\sqrt{2\pi\sigma^2}} \cdot \exp\left\{-\frac{(1-x^T(XX^T+\lambda I)^{-1}x)^2}{2\sigma^2}(y-\hat{y})^2\right\} \tag{29}$$

and the associated regret or the log-normalization factor:

$$\Gamma = \log\kappa = \log\left(\frac{1}{1-x^T(XX^T+\lambda I)^{-1}x}\right) \tag{30}$$

Note that regularization can help in the case where XX^T, the un-normalized correlation matrix of the data is ill conditioned. In the next section we find the "learnable space" for the linear regression problem and observe situations where this regularization is needed.

VI. Learnable Space

In order to understand for which test sample the trained model generalizes well we need to look at the regret expression (24). High regret means that the pNML learner is very far from the genie and therefore we may not trust its predictions. Low regret, on the other hand, means the model is as good as a genie who knows the true test label, and so it is trusted.

Consider the matrix $X_N = [x_1, x_2, \ldots, x_N]$, composed of the training data, and apply the singular value decomposition (SVD) on it, i.e., $X_N = U\Sigma V^T$ with $U \in R^{M\times M}$, Σ is a rectangular diagonal matrix of the singular values and $V \in R^{N\times N}$. The expression $x^T(XX^T)^{-1}x$ can be rewritten as:

$$\begin{aligned} x^T(XX^T)^{-1}x &= x^T\left(\begin{bmatrix} U\Sigma V^T & x\end{bmatrix}\begin{bmatrix} V\Sigma^T U^T \\ x^T\end{bmatrix}\right)^{-1} x \\ &= x^T\left(U\Sigma\Sigma^T U^T + xx^T\right)^{-1} x. \end{aligned} \tag{31}$$

Denote by R_N the empirical correlation matrix of the training:

$$R_N = \frac{1}{N}U\Sigma\Sigma^T U^T = UHU^T \quad R_N^{-1} = UH^{-1}U^T \tag{32}$$

where H is a diagonal matrix with $H_{ii} = \eta_i$, the eigenvalues of R_N. By the matrix inversion lemma, see [17], we have:

$$x^T(XX^T)^{-1}x = x^T\left[\frac{1}{N}R_N^{-1} - \frac{\frac{1}{N^2}R_N^{-1}xx^TR_N^{-1}}{1+\frac{1}{N}x^TR_N^{-1}x}\right]x. \tag{33}$$

Denote $\gamma = x^TR_N^{-1}x$. We can simplify the expression:

$$x^T(XX^T)^{-1}x = \frac{1}{N}\gamma - \frac{\frac{1}{N^2}\gamma^2}{1+\frac{1}{N}\gamma} = \frac{\frac{1}{N}\gamma}{1+\frac{1}{N}\gamma}. \tag{34}$$

Plugging in the regret formula (24):

$$\Gamma = \log\kappa = \log\left(\frac{1}{1-\frac{\frac{1}{N}\gamma}{1+\frac{1}{N}\gamma}}\right) = \log\left(1+\frac{1}{N}\gamma\right). \tag{35}$$

Let u_i be the eigenvectors of the empirical correlation matrix of the training data. Expressing γ by x^Tu_i, the projections of x on u_i:

$$\begin{aligned} \gamma = \begin{bmatrix} x^Tu_1 & \ldots & x^Tu_M\end{bmatrix}\begin{bmatrix} \frac{1}{\eta_1} & \ldots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \ldots & \frac{1}{\eta_M}\end{bmatrix}\begin{bmatrix} u_1^Tx \\ \vdots \\ u_M^Tx\end{bmatrix} \\ = \sum_{i=1}^{M}\frac{\left(x^Tu_i\right)^2}{\eta_i}. \end{aligned} \tag{36}$$

The final regret expression is thus:

$$\Gamma = \log\kappa = \log\left(1+\frac{1}{N}\sum_{i=0}^{M}\frac{\left(x^Tu_i\right)^2}{\eta_i}\right). \tag{37}$$

If the test sample x lies mostly in the subspace spanned by the eigenvectors with large eigenvalues, then the model can generalize well even if $M > N$.

VII. Simulation

In this section we present some simulations that demonstrate the results above. We chose the problem of fitting a polynomial to data, which is a special case of linear regression. The simulation show prediction and generalization capabilities in a variety of polynomial degrees and regularization factors.

In the first experiment we generated 3 random points, t_0, t_1, t_2, uniformly in the interval $[-1, 1]$. These points are the training set and are shown in Figure 1 (top) as red dots. The relation between y and t is given by a polynomial of degree two. Thus, the X matrix of section III is given by:

$$X = \begin{bmatrix} 1 & 1 & 1 \\ t_0 & t_1 & t_2 \\ t_0^2 & t_1^2 & t_2^2\end{bmatrix}. \tag{38}$$

Based on the training we predict a probability for all t values in the interval [-1,1] using (29) with a regularization factor λ of 0, 0.1 and 1.0. It is shown in Figure 1 (top) that without regularization ($\lambda = 0$), the blue curve fits the data exactly. As λ increases the fitted curve becomes less steep but tends to fit less to the training data.

Figure 1 (bottom) shows the regret, given by (24), for the polynomial model from (38) for all $t \in [-1, 1]$ where the training t_i's are marked in red on the x axis. We can see that around the training data the regret is very low in comparison to areas where there are no training data. In addition, models with larger regularization term have lower regret for every point in the interval $[-1, 1]$. For all regularization terms, the regret increases as moving away from the training data.

Next, we simulate the case of fitting polynomials with different degrees. Again, we generated 3 random points in the interval $[-1, 1]$. The matrix X is now:

$$X = \begin{bmatrix} 1 & 1 & 1 \\ t_0 & t_1 & t_2 \\ \vdots & & \\ t_0^{\textit{Poly Deg}} & t_1^{\textit{Poly Deg}} & t_2^{\textit{Poly Deg}}\end{bmatrix}. \tag{39}$$

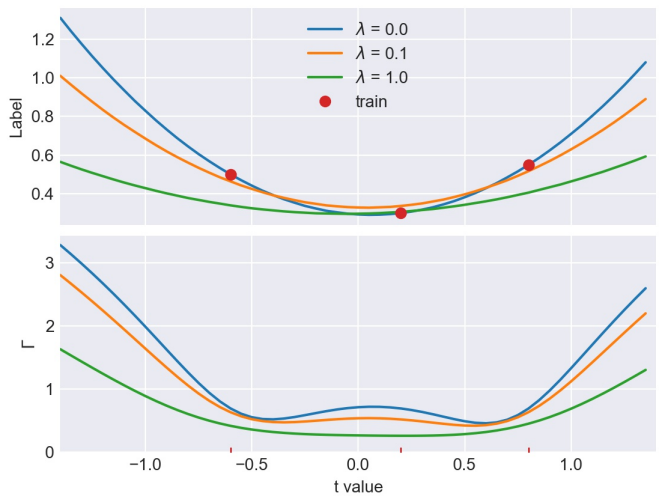

Fig. 1. **Least squares predictor with variety of regularization terms.** (Top) The least squares estimator fitted to the training data (in red) with different values of regularization term. (Bottom) The regret of the pNML learner from (24) on the interval [-1,1]. The training data are marked in red on the x axis.

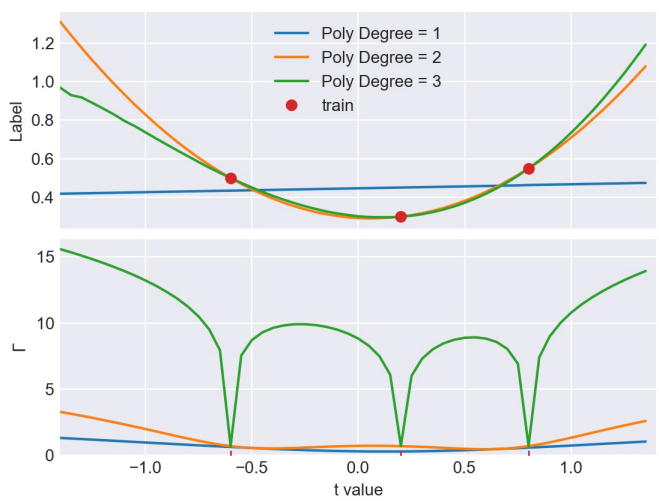

Fig. 2. **Least squares predictor with different polynomial degree.** (Top) pNML least squares predictions with different polynomial degrees. (Bottom) The regret of the pNML learners from (24) on the interval [-1,1]. The training data t values are marked in red on the x axis.

Figure 2 (top) shows the predicted label for every t value in $[-1, 1]$ for the different polynomial degrees. To avoid singularities we used the regularized version with $\lambda = 10^{-6}$. The training set is shown by red dots in the figure. Note that for a polynomial of degree three, the number of parameters is greater than the size of the training set. Nevertheless, the prediction accuracy near the training samples is similar to that of a degree two polynomial. Figure 2 (bottom) shows the regret (or learnability) of the three pNML learners corresponding to model classes of polynomials with the various degrees. All the learners have regret values that are small near the training samples and large as t drifts away from these samples.

VIII. Conclusions

In this paper, we provided an explicit analytical solution of the pNML universal learning scheme and its learnability measure for the linear regression hypothesis class. Interestingly, the predicted universal pNML assignment is Gaussian with a mean that is equal to that of the ERM, but with a variance that increases by a factor κ whose logarithm is the learnability measure Γ. Analyzing Γ we can observe the "learnability space" for this problem. Specifically, if a test sample mostly lies in the subspace spanned by the eigenvectors associated with large eigenvalues of the empirical correlation matrix then the learner can generalize well, even in an over-parameterized case where the regression dimension is larger than the number of training samples. Finally, we provided a simulation of the pNML least squares prediction for polynomial interpolation.

This work suggests a number of potential directions for future work, some are already explored in an accompanying paper [18]. We conjecture that as in linear regression other "over-parameterized" model classes are learnable at least locally, that can be inferred from the pNML solution. This notion is indeed corroborated by the findings in [18].

References

[1] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Siam, 1995, vol. 15.

[2] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[3] N. Merhav and M. Feder, "Universal prediction," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2124–2147, 1998.

[4] Y. Fogel and M. Feder, "Universal supervised learning for individual data," *arXiv preprint arXiv:1812.09520*, 2018.

[5] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[6] Y. M. Shtar'kov, "Universal sequential coding of single messages," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[7] T. Roos and J. Rissanen, "On sequentially normalized maximum likelihood models," 2008.

[8] T. Roos, T. Silander, P. Kontkanen, and P. Myllymaki, "Bayesian network structure learning using factorized nml universal models," in *Information Theory and Applications Workshop, 2008*. IEEE, 2008, pp. 272–276.

[9] V. Vapnik, "Principles of risk minimization for learning theory," in *Advances in neural information processing systems*, 1992, pp. 831–838.

[10] ——, *The nature of statistical learning theory*. Springer science & business media, 2013.

[11] E. D. Sontag, "Vc dimension of neural networks," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 69–96, 1998.

[12] M. H. Hayes, "9.4: Recursive least squares," *Statistical Digital Signal Processing and Modeling*, p. 541, 1996.

[13] K. W. Fornalski, "Applications of the robust bayesian regression analysis," *International Journal of Society Systems Science*, vol. 7, no. 4, pp. 314–333, 2015.

[14] C. Cardinali, "Observation influence diagnostic of a data assimilation system," in *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. II)*. Springer, 2013, pp. 89–110.

[15] H. Trevor, T. Robert, and F. JH, "The elements of statistical learning: data mining, inference, and prediction," 2009.

[16] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, 1970.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Section 2.7. 1 sherman–morrison formula," *Numerical Recipes: The Art of Scientific Computing (3rd ed.). Cambridge University Press, New York*, vol. 1, pp. 55–67, 2007.

[18] K. Bibas, Y. Fogel, and M. Feder, "Deep pnml: Predictive noramalized maximum likelihood for deep neural networks," *Submitted, International Conference on Machine Learning (ICML)*, 2019.