REGRESSION WITH SPARSE APPROXIMATIONS OF DATA

Pardis Noorzad

Department of Computer Engineering and IT
Amirkabir University of Technology
424 Hafez Ave., Tehran, Iran
pardis@aut.ac.ir

Bob L. Sturm

Dept. Architecture, Design and Media Tech.
Aalborg University Copenhagen
A.C. Meyers Vænge 15, DK-2450
Copenahgen SV, Denmark, bst@create.aau.dk

ABSTRACT

We propose sparse approximation weighted regression (SPARROW), a method for local estimation of the regression function that uses sparse approximation with a dictionary of measurements. SPARROW estimates the regression function at a point with a linear combination of a few regressands selected by a sparse approximation of the point in terms of the regressors. We show SPARROW can be considered a variant of k-nearest neighbors regression (k-NNR), and more generally, local polynomial kernel regression. Unlike k-NNR, however, SPARROW can adapt the number of regressors to use based on the sparse approximation process. Our experimental results show the locally constant form of SPARROW performs competitively.

***Index Terms*—** Nonparametric local polynomial regression, multivariate regression, sparse approximation

1. INTRODUCTION

In this paper, we propose and study a new nonparametric method for local multivariate regression — sparse approximation weighted regression (SPARROW) — which employs the sparse approximation of a point in terms of the regressors. A similar nonparametric approach is k-nearest neighbor regression (k-NNR) [1], which assumes that the k regressors nearest to a test point produce similar regressands. Both approaches can be considered variants of local polynomial kernel regression (LPKR) [2], which estimates the regression function at a point by fitting a polynomial at that point.

In addition to local methods like k-NNR and LPKR, considerable research has been aimed at global nonparametric methods, for example, additive models (AMs) [3], and sparse additive models (SpAM) [4]. In AMs, univariate methods are employed to estimate a smooth function of each regressor — in a model consisting of the sum of such univariate component functions — avoiding the need to deal directly with multidimensional inputs. In SpAM, the aim is to reduce the number of component functions of an additive model [4]. Projection pursuit regression (PPR) [5] is an extension to AMs that is able to model a more general class of functions.

Although methods for global parametric and nonparametric regression might minimize the mean error over the entire dataset, it may not provide a good local fit. Local methods like LPKR assume a local parametric model for the data [6]. In LPKR, one estimates the regression function at each point by fitting a Taylor polynomial about that point. This can produce models that are locally constant, locally linear, locally quadratic, etc., based on the order of the polynomial. Central to this procedure is the minimization of a weighted sum of squares error. Typically, the weights are defined by a decreasing function of the distance between two points. SPARROW defines these weights using the sparse approximation of the test point. Implicit in this is the assumption that a test point is better modeled by a sparse linear combination of the regressors than by its proximity to them.

The advantages of data modeling with sparsity constraints are well-documented [7–9], e.g., in uncovering the physiological code of the mammalian primary visual cortex [10], and in producing sparse codes of natural sounds [11], images [12], musical audio [13]. Within the field of supervised learning, sparse representation classification [14] can outperform standard approaches in difficult settings, e.g., speech recognition in noise [15], and face recognition with occlusions, misalignments, and illumination variation [14, 16]. Sparsity has also been applied to variable selection, most notably in the LASSO [17]. In the next sections, we define SPARROW, and show how it is a variant of k-NNR and LPKR. Then we present several experimental results comparing SPARROW with these and other well-known approaches. We make available all our code to reproduce the figures in this paper here: http://imi.aau.dk/∼bst.

2. SPARSE APPROXIMATION WEIGHTED REGRESSION

Consider a dataset (or dictionary) of N observations, $\mathcal{D} := \{(\mathbf{x}_i, y_i)\}_{i\in\Omega}$, where the input $\mathbf{x}_i = [x_{1i}, \ldots, x_{Mi}]^T \in \mathbb{R}^M$ is associated with the output $y_i \in \mathbb{R}$. Let $\Omega := \{1, 2, \ldots, N\}$ index the dictionary. In nonparametric regression, one as-

B. L. Sturm is supported in part by Independent Postdoc Grant 11-105218 from Det Frie Forskningsrd.

sumes $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $f(\mathbf{x})$ is an unknown but smooth function and ϵ_i is some error independent of $\mathbf{x}_i$. Given $\mathcal{D}$ and a point $\mathbf{z}$, SPARROW estimates the regression function $f(\mathbf{z})$ by a linear combination of the outputs

$$\hat{f}(\mathbf{z}) := \sum_{i \in \Omega} l_i(\mathbf{z}, \mathcal{D}) y_i \tag{1}$$

where $l_i(\mathbf{z}, \mathcal{D})$ is the ith *effective weight*, which SPARROW defines as a function of the sparse approximation of $\mathbf{z}$ in $\mathcal{D}$.

Instead of fitting a single model to the entire dataset, as in global parametric and nonparametric regression, SPARROW fits parametric models about each test point $\mathbf{z}$ by using, e.g., a zeroth, first-, or second-order Taylor expansion. We now discuss how SPARROW defines the effective weights in (1) to estimate the regression function at a given point.

2.1. Definition of effective weights

To obtain the local quadratic estimate of the regression function at $\mathbf{z}$, we can approximate $f(\mathbf{x})$ about $\mathbf{z}$ by a Taylor polynomial of degree two

$$f(\mathbf{x}) \approx f(\mathbf{z}) + (\mathbf{x} - \mathbf{z})^T \boldsymbol{\theta}_\mathbf{z} + \frac{1}{2}(\mathbf{x} - \mathbf{z})^T \mathbf{H}_\mathbf{z} (\mathbf{x} - \mathbf{z}) \tag{2}$$

with $\boldsymbol{\theta}_\mathbf{z}$ the gradient of $f(\mathbf{x})$, and $\mathbf{H}_\mathbf{z}$ its Hessian, both evaluated at $\mathbf{z}$. The problem now is to find $f(\mathbf{z}), \theta_\mathbf{z}$ and $\mathbf{H}_\mathbf{z}$, such that we minimize the locally weighted squared error about $\mathbf{z}$ for all measurements in $\mathcal{D}$, i.e.,

$$\min_{f(\mathbf{z}), \boldsymbol{\theta}_\mathbf{z}, \mathbf{H}_\mathbf{z}} \sum_{i \in \Omega} \alpha_i(\mathbf{z}) \Big[y_i - f(\mathbf{z}) - (\mathbf{x}_i - \mathbf{z})^T \boldsymbol{\theta}_\mathbf{z} - \frac{1}{2}(\mathbf{x}_i - \mathbf{z})^T \mathbf{H}_\mathbf{z} (\mathbf{x}_i - \mathbf{z}) \Big]^2 \tag{3}$$

where $\alpha_i(\mathbf{z})$ is the ith *observation weight*, which can be defined in several ways, e.g., by a kernel function [1, 18], or by sparse approximation as done by SPARROW.

Now define the parameter supervector [2]

$$\boldsymbol{\Theta}_\mathbf{z} := \big[f(\mathbf{z}), \boldsymbol{\theta}_\mathbf{z}, \mathrm{vech}(\mathbf{H}_\mathbf{z}) \big]^T \tag{4}$$

where $\mathrm{vech}(\mathbf{H})$ denotes the half-vectorization of the symmetric $M \times M$ matrix, i.e., the $M(M+1)/2$-vector formed by stacking the diagonal and lower triangular entries of $\mathbf{H}_\mathbf{z}$. Define the diagonal matrix $\mathbf{A}_\mathbf{z}$ where its ith diagonal element is the observation weight $\alpha_i(\mathbf{z})$. By defining the matrix

$$\mathbf{X}_\mathbf{z} := \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{z})^T & \mathrm{vech}^T[(\mathbf{x}_1 - \mathbf{z})(\mathbf{x}_1 - \mathbf{z})^T] \\ \vdots & \vdots & \vdots \\ 1 & (\mathbf{x}_N - \mathbf{z})^T & \mathrm{vech}^T[(\mathbf{x}_N - \mathbf{z})(\mathbf{x}_N - \mathbf{z})^T] \end{bmatrix} \tag{5}$$

we can express the minimization in (3) as

$$\min_{\boldsymbol{\Theta}_\mathbf{z}} \Big\| \mathbf{A}_\mathbf{z}^{1/2} \big[\mathbf{y} - \mathbf{X}_\mathbf{z} \boldsymbol{\Theta}_\mathbf{z} \big] \Big\|_2^2 \tag{6}$$

where the regressands vector $\mathbf{y} := [y_1, y_2, \ldots, y_N]^T$. The parameters defined by the least-squares solution is [2]

$$\widehat{\boldsymbol{\Theta}}_\mathbf{z} = \big(\mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{X}_\mathbf{z} \big)^{-1} \mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{y} \tag{7}$$

provided $\mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{X}_\mathbf{z}$ is invertible. Finally, the local quadratic estimate of the regression function at $\mathbf{z}$ is just the first element of $\boldsymbol{\Theta}_\mathbf{z}$, i.e.,

$$\hat{f}(\mathbf{z}) = \mathbf{e}_1^T \big(\mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{X}_\mathbf{z} \big)^{-1} \mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{y} = \sum_{i \in \Omega} \beta_i y_i \tag{8}$$

where $\mathbf{e}_1$ has a one in its first row, and zeros in all others. Hence, we see the ith effective weight in (1) is

$$l_i(\mathbf{z}, \mathcal{D}) = \mathbf{e}_i^T \mathbf{A}_\mathbf{z}^T \mathbf{X}_\mathbf{z} \big(\mathbf{X}_\mathbf{z}^T \mathbf{A}_\mathbf{z} \mathbf{X}_\mathbf{z} \big)^{-1} \mathbf{e}_1 \tag{9}$$

In summary, SPARROW estimates the regression function at a point $\mathbf{z}$ by computing (1) with effective weights given by (9). If we use only the first column of $\mathbf{X}_\mathbf{z}$ in (9), we produce a locally constant estimate of $f(\mathbf{z})$, i.e.,

$$\hat{f}(\mathbf{z}) = (\mathbf{1}^T \mathbf{A}_\mathbf{z} \mathbf{1})^{-1} \mathbf{1}^T \mathbf{A}_\mathbf{z} \mathbf{y} = \frac{\sum_{i \in \Omega} \alpha_i(\mathbf{z}) y_i}{\sum_{k \in \Omega} \alpha_k(\mathbf{z})}. \tag{10}$$

Using $M + 1$ columns of $\mathbf{X}_\mathbf{z}$ produces a locally linear estimate. And using all of $\mathbf{X}_\mathbf{z}$ results in a locally quadratic estimate. Using higher order polynomials as the local parametric model reduces the bias of the estimate [2, 18], but this comes at the price of increased variance and computation time because the number of local parameters to be estimated increases exponentially. Additionally, higher order polynomials do not offer significant improvement over the quadratic model unless one seeks to estimate the gradient and the Hessian, i.e., $\boldsymbol{\theta}_\mathbf{z}$ and $\mathbf{H}_\mathbf{z}$ in (3) [19].

2.2. Definition of observation weights

Since the effective weights in (9) are a function of the observation weights in (3), i.e., $\{\alpha_i(\mathbf{z}) : i \in \Omega\}$, the remaining problem is to define the observation weights. If we define them in the locally constant model (10) by a kernel function, we produce the Nadaraya-Watson regression (NWR) estimate [20]. In this direction, we can define the weights by

$$\alpha_i(\mathbf{z}) := \mathrm{K}\left(\mathrm{S}(\mathbf{z}, \mathbf{x}_i)/h \right) \tag{11}$$

where $\mathrm{K} : \mathbb{R} \mapsto \mathbb{R}_+$ is a kernel function, $h > 0$ is the bandwidth, and $\mathrm{S}(\mathbf{z}, \mathbf{x}_i)$ is the distance

$$\mathrm{S}(\mathbf{z}, \mathbf{x}_i) := (\mathbf{z} - \mathbf{x}_i)^T \mathbf{V}^{-1} (\mathbf{z} - \mathbf{x}_i) \tag{12}$$

where $\mathbf{V}$ is either a diagonal matrix of the unbiased estimates of the variances observed in the dimensions of the regressors in $\mathcal{D}$ (in which case (12) is the scaled Euclidean distance), or the unbiased estimate of the covariance of the regressors (in which case (12) is the Mahalanobis distance).

Dataset	# observations (N)	# attributes (M)	k
Abalone	4,177	8	9
Bodyfat	252	14	4
Housing	506	13	2
MPG	392	7	4

Table 1. Summary of the four datasets we test. The last column indicates the tuned parameter k used in the experiments involving k-NNR and Wk-NNR.

When we define the weights of the locally constant model

$$\alpha_i(\mathbf{z}) := \begin{cases} d(\mathbf{z}, \mathbf{x}_i), & i \in N_k(\mathbf{z}) \subset \Omega \\ 0, & \text{else} \end{cases} \tag{13}$$

where $N_k(\mathbf{z})$ is the index set of the k nearest regressors of $\mathbf{z}$ in $\mathcal{D}$, then (10) produces k-NNR [1]. If $d(\mathbf{z}, \mathbf{x}_i) := 1$, then the bandwidth of the constant kernel from $\mathbf{z}$ is at least as big as the largest distance between pairs of observations and $\mathbf{z}$, i.e., $h \geq \max_{i \in N_k(\mathbf{z})} \mathrm{S}(\mathbf{z}, \mathbf{x}_i)$. In weighted k-NNR (Wk-NNR), we define this weight as the reciprocal of the distance $d(\mathbf{z}, \mathbf{x}_i) := 1/S(\mathbf{z}, \mathbf{x}_i)$.

Contrary to NWR and k-NNR, SPARROW instead defines the observation weights from the sparse approximation of $\mathbf{z}$ in $\mathcal{D}$. First, consider the matrix form of the normalized regressors of the dictionary

$$\mathbf{D} := \left[\frac{\mathbf{x}_1}{\|\mathbf{x}_1\|_2}, \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|_2}, \ldots, \frac{\mathbf{x}_N}{\|\mathbf{x}_N\|_2} \right]. \tag{14}$$

For an input $\mathbf{z}$, SPARROW finds a solution to $\mathbf{z} \approx \mathbf{D}\mathbf{s}$ such that $\mathbf{s} = [s_1, s_2, \ldots, s_N]^T$ has many zero elements. There are a variety of ways to produce sparse approximations (see [8, 9, 21] for reviews). In this work, we use the principle of basis pursuit denoising (BPDN) [7], which poses the problem

$$\min_{\mathbf{s} \in \mathbb{R}^N} \|\mathbf{s}\|_1 \;\text{ subject to }\; \frac{\|\mathbf{z} - \mathbf{D}\mathbf{s}\|_2^2}{\|\mathbf{z}\|_2^2} \leq \epsilon^2 \tag{15}$$

where $\epsilon^2 > 0$ limits the signal to approximation error ratio. Finally, SPARROW defines the ith observation weight using the sparse approximation weights

$$\alpha_i(\mathbf{z}) := \left[\frac{\mathrm{S}(\mathbf{z}, \mathbf{x}_i)}{\min_{j \in \Omega} \mathrm{S}(\mathbf{z}, \mathbf{x}_j)} \right]^{-1} \frac{s_i}{\|\mathbf{z}\|_2} \tag{16}$$

where s_i is the ith element of $\mathbf{s}$. The purpose of the first coefficient is to weight by 1 the regressand of the regressor closest to $\mathbf{z}$; and the purpose of dividing the sparse approximation weight by $\|\mathbf{z}\|_2$ is remove the influence of its length. Thus, as for Wk-NNR, SPARROW weights more heavily an observation closer to the query, but unlike Wk-NNR, only if it has a nonzero coefficient in its sparse approximation with $\mathcal{D}$. When we substitute the weights $\alpha_i(\mathbf{z})$ from (16) into (10) we obtain the constant SPARROW (C-SPARROW) estimate. And when we use these weights in (9), but use only the first $M + 1$ columns of $\mathbf{X}_\mathbf{z}$, (1) produces the linear SPARROW (L-SPARROW) estimate. Using all columns of $\mathbf{X}_\mathbf{z}$ produces the quadratic SPARROW (Q-SPARROW) estimate.

3. EMPIRICAL EVALUATION

We now compare the performance of SPARROW against several other other well-established methods for local regression. In all cases, we use the standardized Euclidean distance in (12). We test NWR and its linear counterpart, local linear kernel regression (LLKR) [1, 18], which solves (7) using the first $M + 1$ columns of $\mathbf{X}_\mathbf{z}$ in (5). For both NWR and LLKR we adopt the Gaussian kernel in (11)

$$\mathrm{K}(x) := \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{17}$$

We also test k-NNR, Wk-NNR [1], for which we tune k by nested cross-validation. For a baseline, we test the global parametric approach of multiple linear regression (MLR) [22], which assumes a linear form of the regression function

$$f(\mathbf{x}) = [1, \mathbf{x}^T]\mathbf{b} \tag{18}$$

and $\mathbf{b}$ is defined to minimize the mean squared error

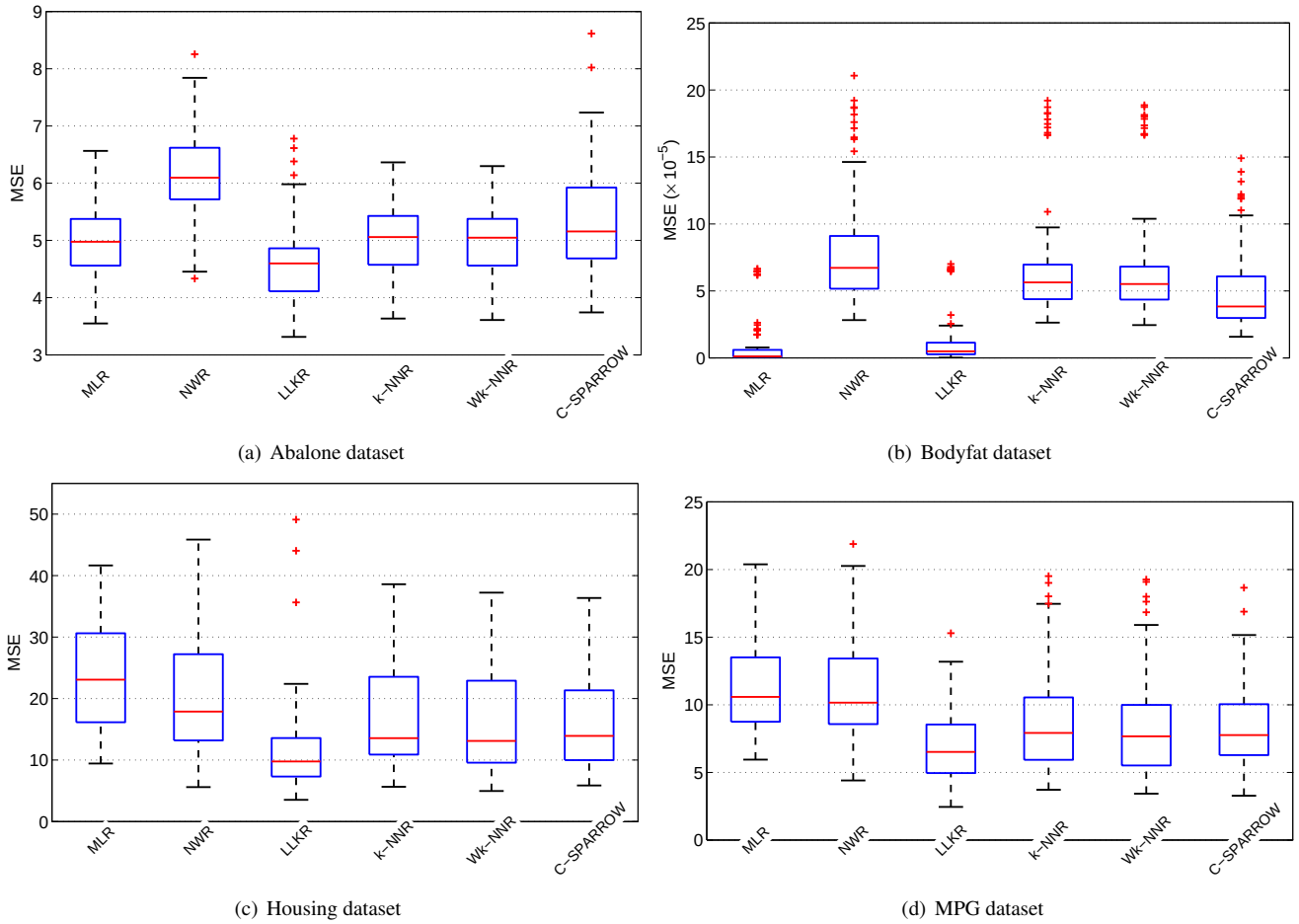
$$\mathbf{b} = \underset{\mathbf{b}' \in \mathbb{R}^{M+1}}{\operatorname{argmin}} \|\mathbf{y} - [\mathbf{1}\ \mathbf{X}^T]\mathbf{b}\|_2^2 \tag{19}$$

where the ith column of $\mathbf{X}$ is $\mathbf{x}_i$. To produce the sparse approximation for a test point in (15), we use the Spectral Projected Gradient Method for ℓ_1-minimization (SPGL1) [23], with at most 20 iterations, and $\epsilon := 10^{-6}$.

We use four different datasets commonly used in regression (see Table 1).[1] Except for Bodyfat, we standardize each dataset such that its dimensions are zero-mean and have the same variance. Figure 1 shows the mean squared error (MSE) estimates of these algorithms from 10 independent trials of 10-fold cross-validation. We see that while MLR performs well for Bodyfat and Abalone, it performs poorly for MPG and Housing. On the other hand, we see that LLKR does extremely well for all datasets. This gain in performance comes with an increase in computation as LLKR must compute (7). Except for Abalone and Housing, we see that C-SPARROW performs nearly the same as k-NNR and Wk-NNR. For Housing, C-SPARROW appears to be almost as good as LLKR. This is surprising since, 1) C-SPARROW makes no assumption of the number of neighbors to be used for each test point, and 2) it is constructing a local constant estimation.

Table 2 shows the performance of L-SPARROW as compared to C-SPARROW. One might expect that L-SPARROW would perform better than C-SPARROW since it is a higher-order model. However, a problem with local polynomial regression for higher order polynomials (i.e., first- and second-order) is that when the input is locally rank deficient, the solutions to (7) become unstable. We resolve the problem by solving a regularized form of the weighted least squares optimization in (3). We use the ℓ_2-norm of the local parameters as

[1] MPG, Abalone and Housing are from http://archive.ics.uci.edu/ml/; Bodyfat is from http://lib.stat.cmu.edu/datasets/.



(a) Abalone dataset

(b) Bodyfat dataset

(c) Housing dataset

(d) MPG dataset

Fig. 1. Boxplots for 10-fold cross-validation estimate of mean squared error (100 independent runs) for four different datasets. Each box delimits 25 to 75 percentiles, and the red line marks median. Extrema are marked by whiskers, and outliers by pluses.

the regularization term thus solving a ridge regression problem [24], i.e., instead of solving (6), we solve

$$\min_{\boldsymbol{\Theta}_{\mathbf{z}},\lambda} \left\| \mathbf{A}_{\mathbf{z}}^{1/2} \left[\mathbf{y} - \mathbf{X}_{\mathbf{z}} \boldsymbol{\Theta}_{\mathbf{z}} \right] \right\|_2^2 + \lambda \|\boldsymbol{\Theta}_{\mathbf{z}}\|_2^2 \qquad (20)$$

where $\lambda \geq 0$ is the *ridge parameter*. For a given λ, the solution becomes [22]

$$\widehat{\boldsymbol{\Theta}}(\mathbf{z}) = \left(\mathbf{X}_{\mathbf{z}}^T \mathbf{A}_{\mathbf{z}} \mathbf{X}_{\mathbf{z}} + \lambda \mathbf{I} \right)^{-1} \mathbf{X}_{\mathbf{z}}^T \mathbf{A}_{\mathbf{z}} \mathbf{y}. \qquad (21)$$

We tune λ in the same way as we do k, described above. Nevertheless, while we see the performance of L-SPARROW improve with respect to using (7), it remains inferior to C-SPARROW.

4. CONCLUSION

In this work, we have proposed an adaptive variation of local polynomial regression methods: NWR, LLKR, k-NNR and Wk-NNR. NWR and LLKR use the entire dataset, and weight the regressand of each regressor by a kernel function.

Dataset	C-SPAR.	L-SPAR. w/ R	L-SPAR.	λ
Abalone	5	16	988	10^{-3}
Bodyfat	5×10^{-5}	35×10^{-5}	960×10^{-5}	10^{-6}
Housing	10	45	4304	10^{-4}
MPG	7	8	6335	10^{-3}

Table 2. A comparison of the MSE estimates obtained on four data sets by 10 trials of 10-fold cross-validation of C-SPARROW and L-SPARROW with and without regularization. The last column denotes the ridge parameter used to obtain the L-SPARROW estimate.

Alternatively, k-NNR and Wk-NNR use the regressands of the k regressors closest to a point, to locally estimate the regression function. With SPARROW, we propose using sparse approximation to adaptively select which regressors to use, and the weights of their regressands to estimate the regression function at a given point. Our experiments show that constant SPARROW can be a competitive regression algorithm. Our future work will analyze the situations where it makes

sense to describe data as a linear combination (including negative weights) of labeled data. Furthermore, one can use other sparse approximation algorithms, such as greedy approaches, which are typically less computationally expensive than convex optimization approaches like BPDN.

5. REFERENCES

[1] W. Härdle and O. Linton, "Applied nonparametric methods," Tech. Rep. 1069, Yale University, 1994.

[2] D. Ruppert and M. P. Wand, "Multivariate locally weighted least squares regression," *The Annals of Statistics*, vol. 22, pp. 1346–1370, 1994.

[3] Andreas Buja, Trevor Hastie, and Robert Tibshirani, "Linear smoothers and additive models," *The Annals of Statistics*, vol. 17, no. 2, pp. 435–555, 1989.

[4] Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman, "Sparse additive models," *Journal of The Royal Statistical Society (Series B)*, vol. 71, no. 5, pp. 1009–1030, 2009.

[5] Jerome H. Friedman and Werner Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, no. 376, pp. 817–823, 1981.

[6] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[7] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, Aug. 1998.

[8] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*, Springer, 2010.

[9] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Elsevier, Amsterdam, 3rd edition, 2009.

[10] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[11] M. S. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, Mar. 2002.

[12] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conf. Computer Vision and Pattern Rec.*, 2009.

[13] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proc. IEEE*, 2009, Accepted for publication.

[14] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[15] Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition.," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 2067–2080, 2011.

[16] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Towards a practical face recognition system: Robust alignment and illumination via sparse representation," To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[17] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[18] T. J. Hastie and C. Loader, "Local regression: Automatic kernel carpentry," *Statistical Science*, vol. 8, no. 2, pp. 120–129, 1993.

[19] D. Ruppert, "Local polynomial regression and its applications in environmental statistics," Tech. Rep., Cornell University, 1996.

[20] E. Nadaraya, "On estimating regression," *Theory of Probability and its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[21] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, 2 edition, 2009.

[23] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[24] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.