Regression Learning with Multiple Noisy Oracles

Kosta Ristovski, Debasish Das, Vladimir Ouzienko, Yuhong Guo, Zoran Obradovic [1]

Abstract. In regression learning, it is often difficult to obtain the true values of the label variables, while multiple sources of noisy estimates of lower quality are readily available. To address this problem, we propose a new Bayesian approach that learns a regression model from data with noisy labels provided by multiple oracles. The proposed method provides closed form solution for model parameters and is applicable to both linear and nonlinear regression problems. In our experiments on synthetic and benchmark datasets this new regression model was consistently more accurate than a model trained with averaged estimates from multiple oracles as labels.

1 INTRODUCTION

In regression learning, it is usually assumed that true labels are readily available to train the learner. However, recent advances in corroborative technology have given rise to situations where the true value of the target is unknown. In such problems, multiple sources or oracles are often available that provide noisy estimates of the target variable. The amount of noise inherent to these estimates may range from slight to high. For example, opinions of human experts about the diameter of a lesion appearing in an x-ray image [11] may be considered slightly noisy, whereas the opinion of a layman is expected to be highly deviant from the actual value. Another example is Amazon's Mechanical Turk [16] (on-line labeling marketplace) where labels for a particular supervised learning task are provided by humans (for a small fee). In this situation human variability in subject matter expertise causes different noise levels in labels. In many other cases obtaining the true value of the label is expensive, whereas obtaining lower quality estimates of the target may be cheap.

In this paper, we address the question of whether it is possible to learn a regression model when provided with multiple noisy labels instead of a single golden standard. We propose a probabilistic Bayesian solution to this question. The Bayesian approach estimates the model parameters as well as the oracle precisions by maximizing the marginal log-posterior of the observed noisy labels and input features. Our approach can be applied to both linear and non-linear regression problems by exploiting the kernel tricks.

The remaining part of this paper consists of the review of related work, followed by presentation of the methodology and derivations, the summary of experimental results, conclusions and discussions of future work.

2 RELATED WORK

Almost all of the previous work related to our problem is devoted to binary classification. In machine learning, this problem first appeared in reference to labeling volcanoes on Venus' surface [15]. In [8] authors provide a preliminary analysis on evaluating classifiers with test data having noisy labels and try to analyze the bounds of the error rate for the classification problem. Their conclusion is that error bounds assuming independence of oracles are not the same if the independence assumption is relaxed. The problem of noisy labels is also considered in a more recent paper [10] where the authors attempted to build a probabilistic model of the classifier in absence of a true label using a latent variable model. There are some recent articles in natural language processing [12] and the computer vision [16] domain where it is shown that using multiple noisy labelers can result in a classifier as good as one trained with labels provided by experts. There are also some theoretical studies [9][13][14] that investigate learning with noisy labels.

A more recent work [4] concerned with multiple noisy oracles provides a simple iterative solution for evaluating the labeler accuracy and fitting a logistic regression model for a binary classification task in absence of the true labels using an exploration-exploitation approach. The assumption in this article is that majority vote is a close approximation of the true label. This idea is further studied by considering a cost-sensitive approach and assuming budgetary constraints [5].

Another important paper [11] published very recently presents an elegant way of solving the problem using a probabilistic approach. This method derives the likelihood observations given the parameters and then uses expectation-maximization to estimate parameter values that maximize the likelihood. Our proposed approach is similar to [11] but with critical differences. First, the proposed method is targeted on regression problems with continuous targets, where one faces new computational challenges that can not be solved by a straightforward extension of the existing methods developed for binary classification. Second, the new method can address both linear and non-linear regressions. Third, unlike [11] we show that closed form solution can be obtained for model weights.

In statistics, Repeated Measurement Regression (RMR) is concerned with making a model for the population observed over time where one measurement is available for each subject at a particular time point [3]. We consider multiple measurements for each point where measurements are of different quality unknown in advance which RMR does not consider. In [6] nonlinear regression is fitted through the means of subjects at each time point which is reasonable for the population but not for our problem.

3 PROBLEM FORMULATION

Let us assume that at the same time for the particular observation vector $\mathbf{x}, K$ oracles provide us with the noisy targets $y_1, \ldots y_K$ where noise is unknown. Each data point is given by $D^{(i)} = \{\mathbf{x}^{(i)}, y_1^{(i)}, ..y_K^{(i)}\}$ where $i = 1, \ldots, N$. Our goal here is to derive the regression function $f(\mathbf{x}, \mathbf{w})$ that maps the inputs $\mathbf{x}$ to target variable y representing estimated golden standard along with the estimate of

[1] Temple University, Philadelphia, USA. email: zoran@ist.temple.edu

precisions of each oracle. It will be shown later that oracle precisions must be estimated very well in order to obtain an accurate estimation of the regression function. Before we start deriving the expressions for *f*(**x, w**) and precisions of oracles, in this section, we introduce the basic assumptions we made and the graphical presentation of the regression problem based on these assumptions.

3.1 Basic Assumptions

First, we assumed the regression to be a linear function in some higher dimensional feature space *F*. We have to choose a fixed mapping from the original space **x** to the feature space *F*, $\phi(\mathbf{x})$: $\mathbf{x} \rightarrow F$. The regression function is given as

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}) \tag{1}$$

With this assumption we avoid the limitation of a linear model and preserve the computational tractability of the model at the same time.

Second, we assume that the regression errors are normally independently distributed with a zero-mean Gaussian distribution. Thus, ground truth label and its probability distribution can be modeled as

$$y = f(\mathbf{x}, \mathbf{w}) + \epsilon, \qquad \epsilon \sim N(0, \sigma_y^2) \tag{2}$$

$$P(y|\mathbf{x}, \mathbf{w}) = N\left(f(\mathbf{x}, \mathbf{w}), \sigma_y^2\right) \tag{3}$$

Moreover, given the true label *y*, we assume all oracles independently provide noisy estimates of the true label. We also assume that the noise intrinsic to each oracle is a Gaussian with different unknown variances.

$$P(y_k|y) = N\left(y, \sigma_k^2\right), k{=}1, \ldots, K \tag{4}$$

Furthermore, we assumed that oracle noise does not depend on input and that oracles provide labels independently from each other. Both these assumptions we intend to relax in our future work.

3.2 Graphical Representation

Upon the assumptions presented above, The Conditional Probability Distributions (CPD) defined by equations (3) and (4) can be represented by a graphical model shown in Figure 1. Here **x** denotes the inputs, the hidden node at the center corresponds to the unobserved true label, and the y_k nodes represent the noisy labels provided by each oracle.

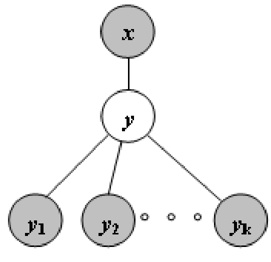


Figure 1. Graphical representation of a regression problem with multiple oracles

4 BAYESIAN ESTIMATE OF LIKELIHOOD

Our goal is to estimate the model parameters (regression weights **w**, oracle precisions $1/\sigma_k^2$ (*k* = 1...*K*) and model noise variance σ_y^2). For convenience, we will denote σ_y^2 by σ_{K+1}^2 and denote *f*(**x, w**) by y_{K+1}. We then use $\boldsymbol{\theta}$ to denote the whole set of parameters that need to be estimated.

Thus the joint probability over oracle labels for a particular instance *i* and for given $\mathbf{x}^{(i)}$ and $\boldsymbol{\theta}$ can be written as

$$\begin{aligned} &P\left(y_1^{(i)}, \ldots, y_K^{(i)} | \mathrm{x}^{(i)}, \theta\right) \\ &= \int_y P\left(y_1^{(i)}, \ldots, y_K^{(i)} | y, \mathrm{x}^{(i)}, \theta\right) P\left(y | \mathrm{x}^{(i)}, \theta\right) dy \end{aligned} \tag{5}$$

Using the independencies between y_k's and **x** given *y* we can rewrite (5) as

$$\begin{aligned} &P\left(y_1^{(i)}, \ldots, y_K^{(i)} | \mathrm{x}^{(i)}, \theta\right) \\ &= \int_y P\left(y_1^{(i)}, \ldots, y_K^{(i)} | y, \theta\right) P\left(y | \mathrm{x}^{(i)}, \theta\right) dy \end{aligned} \tag{6}$$

Again using the independencies among y_k's given *y* we have

$$\begin{aligned} &P\left(y_1^{(i)}, \ldots, y_K^{(i)} | y, \theta\right) = \prod_{k=1}^K P\left(y_k^{(i)} | y, \sigma_1^2, ..\sigma_K^2\right) \\ &= \frac{1}{(2\pi)^{K/2} \prod_{k=1}^K \sigma_k} \exp\left(-\sum_{k=1}^K \frac{\left(y_k^{(i)} - y\right)^2}{2\sigma_k^2}\right) \end{aligned} \tag{7}$$

Substituting (7) and (3) into (6) we obtain

$$\begin{aligned} &P\left(y_1^{(i)}, \ldots, y_K^{(i)} | \mathrm{x}^{(i)}, \theta\right) \\ &= \int_y \frac{1}{(2\pi)^{(K+1)/2} \prod_{k=1}^{K+1} \sigma_k} \exp\left(\begin{array}{c} -\sum\limits_{k=1}^{K} \frac{\left(y_k^{(i)} - y\right)^2}{2\sigma_k^2} \\ -\frac{\left(y - f\left(\mathrm{x}^{(i)}, \mathrm{w}\right)\right)^2}{2\sigma_{K+1}^2} \end{array}\right) dy \\ &= C \int_y \exp\left(-C_1 y^2 + C_{2i} y - C_{3i}\right) dy \\ &= C \sqrt{\frac{\pi}{C_1}} \exp\left(\frac{C_{2i}^2}{4C_1} - C_{3i}\right) \end{aligned} \tag{8}$$

where

$$\begin{aligned} &C = \frac{1}{(2\pi)^{(K+1)/2} \prod_{k=1}^{K+1} \sigma_k} \quad , \quad C_1 = \sum_{k=1}^{K+1} \frac{1}{2\sigma_k^2} \\ &C_{2i} = \sum_{k=1}^{K+1} \frac{y_k^{(i)}}{\sigma_k^2} \quad , \quad C_{3i} = \sum_{k=1}^{K+1} \frac{\left(y_k^{(i)}\right)^2}{2\sigma_k^2} \end{aligned} \tag{9}$$

Now the joint probability over all *N* instances is given by

$$P\left(y_1, \ldots, y_{\mathrm{K}} | X, \theta\right) = \prod_{i=1}^N C \sqrt{\frac{\pi}{C_1}} \exp\left(\frac{C_{2i}^2}{4C_1} - C_{3i}\right) \tag{10}$$

where **X** is the input matrix of all instances given by

$$\mathbf{X} = \begin{bmatrix} 1 & \ldots & 1 \\ x_{11} & & x_{1N} \\ \vdots & & \vdots \\ x_{M1} & \ldots & x_{MN} \end{bmatrix}_{((M+1) \times N)}$$

and M is the number of features in the input vector X. The initial row of 1's is added to accommodate the bias term w_0.

Then the conditional log-likelihood can be written as

$$\begin{aligned} l\left(\mathrm{y}_1, \ldots, \mathrm{y}_{\mathrm{K}}, \mathrm{X}, \theta\right) &= \log P\left(\mathrm{y}_1, \ldots, \mathrm{y}_{\mathrm{K}} | \mathrm{X}, \theta\right) \\ &= N \log C + \frac{N}{2} \log \pi \\ &\quad -\frac{1}{2} N \log C_1 + \sum_{i=1}^{N} \frac{C_{2i}^2}{4C_1} \\ &\quad - \sum_{i=1}^{N} C_{3i} \end{aligned} \tag{11}$$

We further consider a regularization term $-\boldsymbol{\lambda}\mathbf{w}^T\mathbf{w}/2$ which corresponds to isotropic Gaussian prior over **w** [2]. In that way we obtain log-posterior

$$\begin{array}{l}
l'\left(\mathrm{y}_1,\ldots,\mathrm{y}_\mathrm{K},\mathrm{X},\theta\right) \\
\qquad\begin{array}{cl}
= & \log P\left(\theta|\mathrm{X},\mathrm{y}_1,\ldots,\mathrm{y}_\mathrm{K}\right) \\
\propto & \log P\left(\mathrm{y}_1,\ldots,\mathrm{y}_\mathrm{K}|\mathrm{X},\theta\right) \\
 & +\log P(\theta) \\
= & N\log C-\frac{1}{2}N\log C_1+\sum\limits_{i=1}^{N}\frac{C_{2i}^2}{4C_1} \\
 & -\sum\limits_{i=1}^{N}C_{3i}-\frac{\lambda}{2}\mathrm{w}^T\mathrm{w}+const.
\end{array}
\end{array} \tag{12}$$

5 MAXIMUM A-POSTERIORI PARAMETER ESTIMATES

In order to find the Maximum A-Posteriori (MAP) estimates of model weights and oracle precisions to maximize the log-posterior given in equation (12), We first derive the gradients of log-posterior *l'* with respect to 1/$\boldsymbol{\sigma}_k^2$ and **w**, respectively. Let us denote

$$\mathbf{Y}=\begin{bmatrix} y_1^{(1)} & \cdots & y_K^{(1)} & y_{K+1}^{(1)} \\ \vdots & & \vdots & \vdots \\ y_1^{(N)} & \cdots & y_K^{(N)} & y_{K+1}^{(N)} \end{bmatrix}$$

$$\mathbf{S}=\begin{bmatrix} \frac{1}{\sigma_1^2} & \cdots & \frac{1}{\sigma_K^2} & \frac{1}{\sigma_{K+1}^2} \end{bmatrix}^T$$

Then

$$\begin{array}{rcl}
\frac{\partial l'(y_1,\ldots,y_k,X,\theta)}{\partial\left(1/\sigma_k^2\right)} & = & \frac{N\sigma_k^2}{2}-\frac{N}{4C_1} \\
 & & +\frac{S^TY^T\left(2\mathrm{y_kC_1}-\frac{1}{2}\mathrm{YS}\right)}{4C_1^2} \\
 & & -\frac{1}{2}\mathbf{y}_k^T\mathbf{y}_k
\end{array} \tag{13}$$

where $\mathbf{y}_k$ is the *k*-th column of **Y**. By setting the derivative equal to zero we obtain

$$\sigma_k^2=\frac{1}{2C_1}-\frac{S^TY^T\left(2\mathrm{y_kC_1}-\frac{1}{2}\mathrm{YS}\right)}{4C_1^2}+\frac{1}{N}\mathbf{y}_k^T\mathbf{y}_k \tag{14}$$

Now, to estimate **w**, we choose to minimize *l"* = —*l'* instead of maximizing *l'* (see equation (12)) for convenience. After some rearrangements *l"* can be expressed in the following convenient form.

$$l''=A\left(\frac{1}{2}\sum_{i-1}^{N}\left\{\mathrm{w}^T\phi(\mathrm{x}^{(i)})-t_i\right\}^2+\frac{L}{2}\mathrm{w}^T\mathrm{w}\right)+B \tag{15}$$

where

$$L=\frac{2\lambda C_1\sigma_y^4}{2C_1\sigma_y^2-1},\ t_i=\sum_{k=1}^{K}\frac{y_k^i}{\sigma_k^2}\Bigg/\sum_{k=1}^{K}\frac{1}{\sigma_k^2} \tag{16}$$

and A and B are constants independent of **w**. Again minimizing *l"* with respect to **w** is equivalent to minimizing

$$J(\mathrm{w})=\frac{1}{2}\sum_{i-1}^{N}\left\{\mathrm{w}^T\phi(\mathrm{x}^{(i)})-t_i\right\}^2+\frac{L}{2}\mathrm{w}^T\mathrm{w} \tag{17}$$

The expression in (17) is very similar to regularized sum of squares with the target variable t_i being the sum of all labels weighted by respective oracle precisions.

After differentiating *J*(**w**) with respect to **w**, equating to zero, and rearranging we found expression for **w** in the form

$$\begin{array}{l}
\mathrm{w}=\frac{1}{L}\left[\sum_{i=1}^{N}\left(t_i-\mathrm{w}^T\phi(\mathrm{x}^{(i)})\right)\phi(\mathrm{x}^{(i)})\right] \\
=\sum_{i=1}^{N}a_i\phi(x^{(i)})=\mathbf{\Phi}^T\mathbf{a}
\end{array} \tag{18}$$

Here, $\mathbf{\Phi}$ is the design matrix, whose *i*-th row is given by $\phi\ (\mathbf{x}^{(i)})^T$, and **a** is given by $\mathbf{a}=[a_1\ldots a_N]^T$ where

$$a_\mathrm{i}\ =\ \frac{1}{\mathbf{L}}(t_i-\mathrm{w}^T\phi(\mathrm{x}^{(i)})) \tag{19}$$

Substituting value of **w** from (18) into the expression of *J*(**w**) in (17) we obtain

$$\begin{array}{rcl}
J(\mathrm{a}) & = & \frac{1}{2}\mathbf{a}^T\mathbf{\Phi\Phi}^T\mathbf{\Phi\Phi}^T\mathbf{a}-\mathbf{a}^T\mathbf{\Phi\Phi}^T\mathbf{t} \\
 & & +\frac{1}{2}\mathbf{t}^T\mathbf{t}+\frac{L}{2}\mathbf{a}^T\mathbf{\Phi\Phi}^T\mathbf{a}
\end{array} \tag{20}$$

where **t** = $[t_1\ldots.t_N]^T$. Now we introduce the *Gram Kernel matrix* defined as **K**= $\mathbf{\Phi\Phi}^T$.This is an *N* × *N* symmetric matrix with elements

$$K_{ij}=\phi(\mathrm{x}^{(i)})^T\phi(\mathrm{x}^{(j)})=k(\mathrm{x}^{(i)},\mathrm{x}^{(j)}) \tag{21}$$

where $k(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$ is determined by a *kernel function.* In terms of **K**, the expression for *J*(**a**) becomes

$$J(\mathrm{a})=\frac{1}{2}\mathbf{a}^T\mathbf{KKa}-\mathbf{a}^T\mathbf{Kt}+\frac{1}{2}\mathbf{t}^T\mathbf{t}+\frac{L}{2}\mathbf{a}^T\mathbf{Ka} \tag{22}$$

Setting the gradient of *J*(**a**) with respect to **a** to zero, we obtain

$$\mathbf{a}=(\mathbf{K}+L\mathbf{I}_N)^{-1}\mathbf{t} \tag{23}$$

So, if we substitute this value of **a** into (1), we get

$$\begin{array}{rcl}
f(\mathrm{x}^{(i)},\mathrm{w}) & = & \mathrm{w}^T\phi(\mathrm{x}^{(i)}) \\
 & = & \mathbf{a}^T\mathbf{\Phi}\phi(\mathrm{x}^{(i)}) \\
 & = & \mathbf{k}(\mathrm{x}^{(i)})^T(\mathbf{K}+L\mathbf{I}_N)^{-1}\mathbf{t}
\end{array} \tag{24}$$

And for a new input **x**,

$$y(\mathrm{x})=\mathbf{k}(\mathrm{x})^T(\mathbf{K}+L\mathbf{I}_N)^{-1}\mathbf{t} \tag{25}$$

where **k**(**x**) is the vector with elements $k_i(\mathbf{x})\ =\ k(\mathbf{x}^{(i)},\mathbf{x})$. The kernel function we used in this work is the Gaussian Kernel given by,

$$k(\mathrm{x}^{(\mathrm{i})},\mathrm{x}^{(\mathrm{j})})\ =\ \mathrm{e}^{-\frac{\left(\mathrm{x}^{(\mathrm{i})}-\mathrm{x}^{(\mathrm{j})}\right)^2}{2\sigma^2}} \tag{26}$$

In our experiments, we set the regularization parameter $\boldsymbol{\lambda}$ and the kernel parameter $\boldsymbol{\sigma}$ experimentally.

The closed form solutions we derived for $\boldsymbol{\sigma}_k^2$ and *f*$(\mathbf{x}^{(i)},\mathbf{w})$ are interdependent. From equation (14), we see that the value of $\boldsymbol{\sigma}_k^2$ depends on itself (through **S**) and on *f*$(\mathbf{x}^{(i)},\mathbf{w})$ (through **Y**). Also, From equation (24), we see that the values of *f*(**x**(i), **w**) depend on $\boldsymbol{\sigma}_k^2$ (through *L* and **t**). Finding independent solutions analytically is difficult. So, we use an iterative optimization method to find the values of $\boldsymbol{\sigma}_k^2$ and *f*$(\mathbf{x}^{(i)},\mathbf{w})$. To achieve this objective, we start from a reasonable guess of $\boldsymbol{\sigma}_k^2$ and use equation (24) to determine *f*$(\mathbf{x}^{(i)},\mathbf{w})$. Then we use the obtained values of *f*$(\mathbf{x}^{(i)},\mathbf{w})$ to recompute $\boldsymbol{\sigma}_k^2$ using equation (14). We proceed iteratively until convergence, i.e. until there is no more significant change in values of $\boldsymbol{\sigma}_k^2$.

6 SPEEDUP FOR LINEAR REGRESSION

Nonlinear regression using the Kernel trick as described in the previous section is also applicable to linear regression problems. In these situations we can use linear kernels, but this approach has some disadvantages. As evident from equation (23) and equation (24) the proposed approach for estimating model weights includes inversion of a $N \times N$ matrix which would increase the time complexity for solving linear regression problems. Thus we solve linear regression problem by setting

$$f(\mathrm{x}, \mathrm{w}) = \mathrm{w}^T \mathrm{x}, \; \mathrm{x} = [1 \; x_1 \; \ldots \; \mathrm{x}_M]^T \tag{27}$$

In such case the expression for the oracle accuracies is the same while expression for the weights can be estimated as

$$\mathrm{w} = \left(\mathrm{X}\mathrm{X}^T\right)^{-1} \mathrm{X}\mathrm{t} \tag{28}$$

In this case we need to invert an $(M+1) \times (M+1)$ matrix. In cases where $M \ll N$ (which is very common in reality) equation (28) will provide a much faster solution than equation (23).

7 EXPERIMENTS

7.1 Data and Experimental Setup

We have tested the performance of our method on both artificial and six benchmark datasets. Five of the test datasets chosen from UCI repository [1] have nonlinear behaviours, while the *houseprice* dataset chosen from statistics [7] is a linear regression problem. The datasets and their properties are summarized in Table 1. A linear synthetic dataset is used to make sure that experimental data really has linear behaviour.

Target values that appear in the datasets have been considered as ground truth. In order to validate the proposed algorithm we simulate multiple noisy oracles by adding a different amount of Gaussian noise to actual labels. Based on these simulated targets we can simultaneously learn regression model and estimate oracle precisions. Since a priori information about oracles is unknown we treat them equally important in the baseline method. Therefore, the regression model trained on the average of labels was our baseline. For the purpose of better insight in quality of the proposed method we will report prediction accuracies of the models trained on the ground truth as well as on the each oracle separately.

If all oracles are experts (small noise level), then it would be expected that our method performs almost the same as the baseline which makes the case rather uninteresting. In our experiments three oracles were used to assign three target values to each instance and for each dataset, and the following three scenarios were considered:

- Experiment I: one oracle is expert (small noise) and other two are inexperienced (larger noise level).
- Experiment II: all oracles are inexperienced.
- Experiment III: two oracles are inexperienced and one is totally random (huge level of noise)

Accuracies of the models are reported using coefficient of determination (R^2) defined as:

$$R^2 = 1 - \frac{\sum_i \left(y^{(i)} - f(\mathrm{x}^{(i)}, \mathrm{w})\right)^2}{\sum_i \left(y^{(i)} - \bar{y}\right)^2} \tag{29}$$

Values of R-square closer to 1 are better. In all experiments R^2 values were calculated using predictions of particular method

Table 1. Benchmark dataset description

Dataset	No. instances	No. features
Houseprice	107	4
Automobile	159	18
Breast Cancer W.P	194	34
Auto MPG	392	7
Housing	506	13
Concrete	1030	8

Table 2. Accuracy (R^2) on *Automobile* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.81	0.80	0.71	0.66	0.78	**0.80**
Exp II	0.80	0.78	0.70	0.65	0.77	**0.78**
Exp III	0.81	0.78	0.71	-1.29	0.52	**0.79**

Table 3. Accuracy (R^2) on *Breast Cancer W.P.* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	1.00	0.98	0.77	0.54	0.90	**0.98**
Exp II	1.00	0.86	0.76	0.55	0.90	**0.93**
Exp III	1.00	0.86	0.76	-0.31	0.78	**0.93**

Table 4. Accuracy (R^2) on *Auto MPG* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.87	0.87	0.75	0.66	0.82	**0.87**
Exp II	0.87	0.81	0.74	0.65	0.81	**0.84**
Exp III	0.87	0.81	0.75	-0.92	0.58	**0.84**

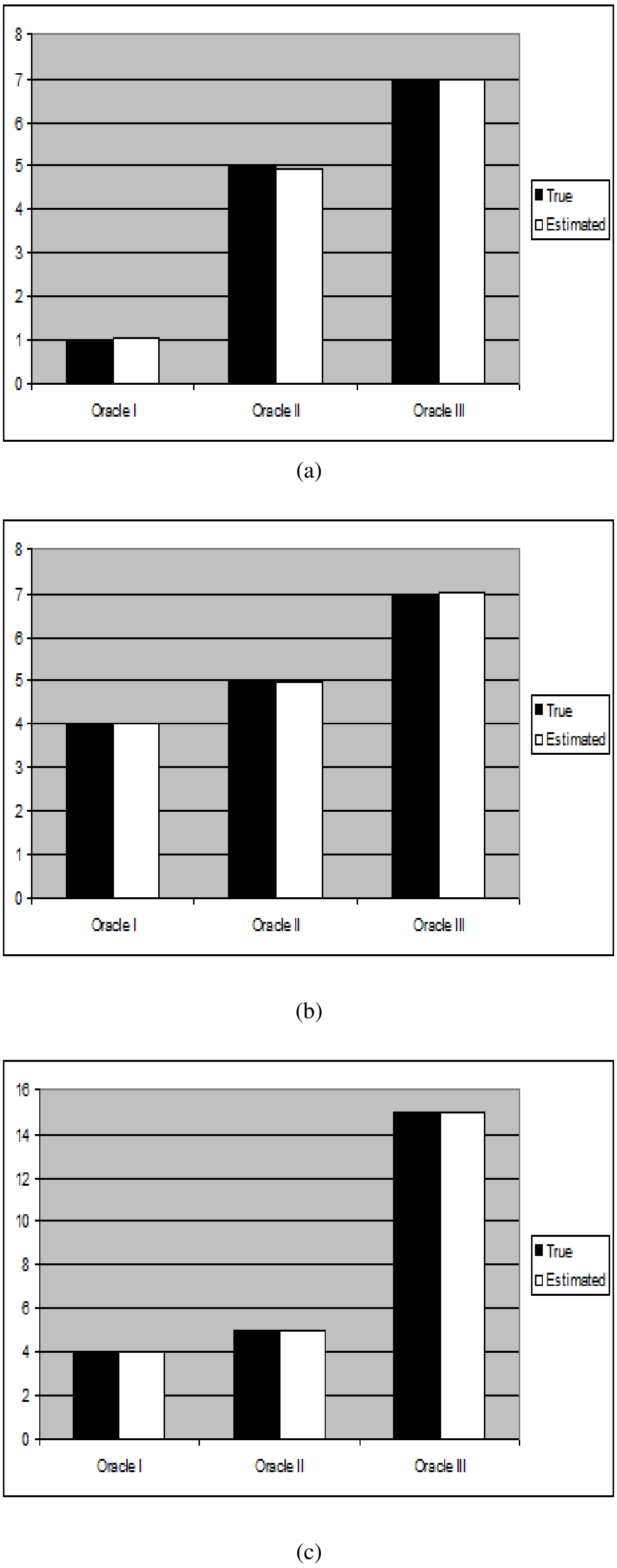
Table 5. Accuracy (R^2) on *Housing* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.87	0.86	0.80	0.73	0.83	**0.86**
Exp II	0.86	0.79	0.79	0.73	0.82	**0.85**
Exp III	0.87	0.80	0.80	0.26	0.75	**0.85**

Table 6. Accuracy (R^2) on *Concrete C.P.* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.87	0.86	0.73	0.66	0.82	**0.86**
Exp II	0.87	0.83	0.72	0.66	0.81	**0.84**
Exp III	0.87	0.82	0.72	-5.29	-0.03	**0.83**

$f(\mathbf{x}, \mathbf{w})$ and the ground truth y. Here, each data set is partitioned into 70%/30% train/test sets and the average results on test sets over 200 runs are reported.



(a)

(b)

(c)

Figure 2. Results on *Housing* dataset for experiments 1,2 and 3 are shown in panels a,b and c respectively.

7.2 Nonlinear Regression Results

Results for experiments performed on five datasets with nonlinear behaviour are presented in Tables 2-6. We can notice that for a particular dataset accuracies of the true model vary slightly in multiple experiments. This occurs due to randomness in choosing training and test sets. However, in all experiments the R-square value for the proposed method was better than for the baseline method. In presence of an almost random oracle (3^{rd} experiment) on all five datasets our method was a lot better than the baseline method. A predictor trained on true labels performs the best as expected. Accuracies for this predictor and the proposed method were slightly different even when the proposed method was trained on very noisy labels used in the third experiment. Moreover, the proposed method performed the same or better than the predictor trained on labels from the best oracle. In presence of one expert oracle and two inexperienced oracles (experiment I) a huge weight was assigned to the expert by our method appropriately. The obtained results show that in those situations accuracy of our model was as good as the accuracy of the expert. On the other hand, when learning without an expert among oracles (experiments II and III) our model took into account information provided by all oracles, which caused the accuracy of our model to be better than the accuracy of the best oracle.

Estimates of accuracies for three oracles obtained by our algorithm over experiments 1-3 on housing data are shown in Figure 2. These results are reported in the form of standard deviation of noise in labels and are compared to the corresponding true values. As evident from Figure 2, in experiments on *housing* data estimated standard deviations of noise level were almost the same as the true values. Essentially identical findings were observed in the corresponding experiments on the remaining datasets (figures omitted for lack of space).

7.3 Linear Regression Results

For evaluation of linear regression, we have constructed a synthetic linear dataset using the following equation to generate true targets

$$y = x_1 - 3.5x_2 + 4x_3 + 5x_4 + 2x_5 + N(0, \sigma_y^2) \tag{30}$$

where σ_y^2 represents model variance introduced in order to avoid perfect linearity. In our experiments value of σ_y^2 was set to one while values of the features (x values) were sampled from Normal distribution.

Experimental results for synthetic and linear dataset are presented in Tables 7 and 8. They show the same behaviour as in nonlinear case. Linear model is also able to estimate accuracies of each oracle quite well which is shown in Table 9.

Table 7. Accuracy (R^2) on *Synthetic* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.98	0.98	0.84	0.76	0.94	**0.98**
Exp II	0.98	0.92	0.86	0.79	0.94	**0.95**
Exp III	0.98	0.92	0.86	-1.93	0.64	**0.94**

Table 8. Accuracy (R^2) on *Housprice* dataset

	True	Orac.I	Orac.II	Orac.III	Baseline	Prop.
Exp I	0.76	0.76	0.67	0.54	0.72	**0.76**
Exp II	0.77	0.71	0.69	0.54	0.73	**0.74**
Exp III	0.76	0.71	0.65	-1.84	0.44	**0.72**

Table 9. True and estimated oracle precisions for *Housprice* dataset represented as standard deviation of noise in labels

	Oracle I		Oracle II		Oracle III	
	True	Est.	True	Est.	True	Est.
Exp I	50	163	400	424	600	612
Exp II	300	326	400	423	600	610
Exp III	300	329	400	416	2000	1984

8 CONCLUSION AND FUTURE WORK

The kernel method for nonlinear regression can be time-consuming in applications to datasets in which the number of instances is much larger than the number of features. Because of that, we also offered a special solution for linear cases. The use of kernels requires regularization and kernel parameters to be properly adjusted.Therefore, development of a method that uses a neural network as a nonlinear model is a part of the future work. The assumption that an oracle maintains uniform precision over all instances will be relaxed in a follow up article where a generalized model with input-dependent oracle accuracy will be considered.

REFERENCES

[1] A. Asuncion and D.J. Newman. UCI machine learning repository. University of California, School of Information and Computer Science. www.ics.uci.edu/ mlearn/MLRepository.html.

[2] C.M. Bishop and C.S. Qazaz, ‘Regression with input-dependent noise: A bayesian treatment’, *Advances in Neural Information Processing Systems*, **9**, 347–353, (1997).

[3] M. Davidian and D.M. Giltinan, ‘Nonlinear models for repeated measurement data. an overview and update.’, *J Agr Biol Envir St.*, (2003).

[4] P. Donmez and J. G. Carbonell, ‘Proactive learning: cost-sensitive active learning with multiple imperfect oracles’, in *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, Napa Valley, California, USA, (2009).

[5] P. Donmez, J. G. Carbonell, and J. Schneider, ‘Efficiently learning the accuracy of labeling sources for selective sampling’, in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Paris, USA, (2009).

[6] J.D. Hart and T.E. Wehrly, ‘Kernel regression estimation using repeated measurements data.’, *J Am. Stat. Assoc.*, (1986).

[7] R. M. Heiberger and Holland B. House price dataset. In Statistical Analysis and Data Display. http://astro.ocis.temple.edu/ rmh/HH hh052509/datasets/houseprice.dat.

[8] C. P. Lam and D. G. Stork, ‘Evaluating classifiers by means of test data with noisy labels’, in *Proceedings of the International Joint Conference on AI (IJCAI)*, Acapulco, Mexico, (2003).

[9] G. Lugosi, ‘Learning with an unreliable teacher’, *Pattern Recognition*, **25**, 79–87, (1992).

[10] C. Pal, G. Mann, and R. Minerich, ‘Putting semantic information extraction on the map: noisy label models for fact extraction’, in *Proceedings of the Workshop on Information Integration on the Web at AAAI*, Vancouver, British Columbia, Canada, (2007).

[11] V.C. Raykar, S. Yu, L.H. Zhao, A. Jerebko, and C. Florin, ‘Supervised learning from multiple experts: whom to trust when everyone lies a bit’, in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, Montreal, Canada, (2009).

[12] V. S. Sheng, F. Provost, and P. G. Ipeirotis, ‘Get another label? improving data quality and data mining using multiple, noisy labelers’, in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, (2008).

[13] B. W. Silverman, ‘Some asymptotic properties of the probabilistic teacher’, *IEEE Transactions on Information Theory*, **26**, 246–249, (1980).

[14] P. Smyth, ‘Learning with probabilistic supervision’, *Computational Learning Theory and Natural Learning Systems*, **3**, 163–182, (1995).

[15] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, ‘Inferring ground truth from subjective labeling of venus images’, *Advances in Neural Information Processing Systems*, **7**, 1085–1092, (1995).

[16] A. Sorokin and D. Forsyth, ‘Utility data annotation with amazon mechanical turk.’, in *Proceedings of the First IEEE Workshop on Internet Vision at IEEE Conference on Computer Vision and Patter Recognition*, Anchorage, Alaska, USA, (2008).