

Online Regression with Controlled Label Noise Rate

Edward Moroshko and Koby Crammer

Department of Electrical Engineering
The Technion, Haifa, Israel
edward.moroshko@gmail.com, koby@ee.technion.ac.il

Abstract. Many online regression (and adaptive filtering) algorithms are linear, use additive update and designed for the noise-free setting. We consider the practical setting where the algorithm’s feedback is noisy, rather than a clean label. We propose a new family of algorithms which modifies the learning rate based on the noise-variance of the feedback (labels), by shrinking both inputs and feedbacks, based on the amount of noise per input instance. We consider both settings, where the noise is either given or estimated. Empirical study with both synthetic and real-world speech data shows that our algorithms improve the overall performance of the regressor, even when there is no additional explicit information (i.e. amount of noise). We also consider a more general setting where an algorithm can sample more than single (noisy) label, yet there is a total (or average) budget for the feedback. We propose a few strategies how to effectively spend the given budget, which are based on noise-variance estimation and our shrinkage rule. We show empirically that our approach outperforms other naive approaches.

Keywords: Online learning, Regression, Adaptive filtering, Label-Noise

1 Introduction

Many online regression algorithms (aka adaptive filters in the signal processing community [19]) are designed for a noise-free setting, or designed for the general case where there is some noise, yet only global properties of it are known, and no additional (or local) per-input knowledge is known or taken into consideration. However, there are practical applications where it is possible to estimate the noise for each input example, or the noise properties may be known. For example, in adaptive filtering systems designed for acoustic echo cancellation, the label noise variance can be estimated during silences [18]. In channel equalization systems that used in digital receivers, the noise variance per-input instance may be given as side information to the equalizer from other channel. In other cases where privacy is an issue, data may be “sanitized”, which corresponds to perturbing data items with some noise [6].

We propose new online regression algorithms for the case that the variance of the noise per input-instance is known. Our algorithms are similar to the

Widrow-Hoff algorithm [24] and are derived from an optimization of an objective capturing the change of a model and loss (or cost) on the current input [13]. We also derive a few simple strategies to estimate the noise variance, and yield algorithms that do not assume any knowledge about the statistical properties of the noise. All of our algorithms modify the learning rate based on the noise (directly or indirectly). We evaluate our algorithms in two synthetic settings and also with a real-world speech filtering task, and show that our algorithms perform well, especially when the noise is not stationary. We then extend the setting to the case when a learning algorithm is allowed to measure (or query) more than single (noisy) feedback label, and propose a label sampling rule that outperforms other possible approaches.

Most of previous work on label noise consider the *classification* setting, see two previous works [16, 14] and the references therein. Cesa-Bianchi et al [5] considered online learning with noisy data and square loss. In their setting the features and the labels are noisy, and they analyzed the effect of noise on the stochastic gradient descent (SGD) algorithm. Their regret bound scales with the amount of noise. In addition, for the case where the feature’s noise covariance is known they suggested an adaptation of the SGD algorithm. We focus on *label* noise and propose another modification. Note that in the setting of Cesa-Bianchi et al [5] there is a global known bound on the noise variance for all examples, however we assume that per-example noise variance is known (or estimated).

The rest of the paper is organized as follows. In Section 2 we formally introduce the setting of the problem. Then, in Section 3 we derive a scaling rule to deal with label noise. In Section 4 we perform extensive empirical evaluation that demonstrate the usefulness of our approach, even when the noise variance is unknown. We summarize in Section 5 and outline some future research.

2 Problem Setting

We consider online linear regression with noisy feedback (or measurement). On each round t , a learning algorithm observes an input instance $x_t \in \mathbb{R}^d$ and outputs a prediction $\hat{y}_t \in \mathbb{R}$. After an algorithm outputs a prediction, it has access to a *noisy* unbiased version \tilde{y}_t of the true label $y_t \in \mathbb{R}$, i.e. $\mathbb{E}\tilde{y}_t = y_t$ where the noise’s variance is finite, $\mathbb{E}(\tilde{y}_t - y_t)^2 \doteq v_t$. We first assume that v_t is known to the learning algorithm. Later, in Section 4 we consider also the case where v_t is unknown. After observing the noisy label \tilde{y}_t an algorithm may update its prediction rule, and then proceeds to the next round.

The loss of an algorithm on round t is measured by the expected square loss $\ell_t(\text{alg}) = \mathbb{E}(\hat{y}_t - y_t)^2$ where the expectation is with respect to labels noise distribution and any randomization in an algorithm. The total loss after T rounds is $L_T(\text{alg}) = \sum_{t=1}^T \ell_t(\text{alg})$. Note that an algorithm is evaluated against the true labels y_t and not the noisy labels \tilde{y}_t .

3 Online Regression with Label Noise

A standard paradigm for deriving learning algorithms is minimizing a loss function under a regularization constraint. Equivalently, one can minimize an unconstrained objective which is a sum of the loss and a regularization term. In online learning, a common practice (e.g. [13, 9]) to derive an update rule for the learned model parameters \mathcal{M} at time step t is to minimize the function,

$$C(\mathcal{M}) = \text{dis}(\mathcal{M}, \mathcal{M}_{t-1}) + \eta \ell(\mathcal{M}, (x_t, y_t)) , \quad (1)$$

where $\eta > 0$ is a tradeoff parameter between being correct on the current input and being conservative, i.e. not being far from the previous model [13]. Here, $\text{dis}(\mathcal{M}, \mathcal{M}_{t-1})$ is a dissimilarity (distance) measure between a new model \mathcal{M} and the previously learned model \mathcal{M}_{t-1} , and $\ell(\mathcal{M}, (x_t, y_t))$ is the loss of the new model \mathcal{M} on the example (x_t, y_t) . First-order algorithms [13, 8, 4], such as the Widrow-Hoff algorithm [24] (known also as LMS or its variant NLMS [19]), often employ a weight vector $\mathcal{M} = \{w\}$, while second-order algorithms [20, 15, 9–11, 23] model also second-order covariance-like information, maintaining both a vector w and a PSD matrix Σ , i.e. $\mathcal{M} = \{w, \Sigma\}$. Kivinen and Warmuth [13] derived first-order algorithms for regression by minimizing (1) with the squared loss. For the case when dis is the euclidean distance, minimizing (1) results in a simple additive update rule for w . For the case when dis is the relative entropy function, minimizing (1) results in the exponentiated gradient (EG) algorithm. This paradigm was later applied to classification, where the hinge loss was used rather than the square loss (e.g. first-order passive-aggressive algorithms [8] and second-order confidence weighted algorithms [9]).

We describe a framework for online regression with label noise. In order to deal with label noise we shrink each noisy example (x_t, \tilde{y}_t) by a factor $\sqrt{\alpha_t}$, where $\alpha_t \in [0, 1]$. (The motivation for using a square-root will be clear shortly.) Intuitively, when the noise variance v_t is large for an example (very noisy label), we like α_t to be small so that this example will have a small affect on the learned model \mathcal{M} . Our main goal is to derive rules for $\alpha_t = \alpha_t(v_t)$. As one cannot directly minimize (1) (because the clean y_t is unknown), we approximate the objective (1) to be

$$\tilde{C}(\mathcal{M}; \alpha_t) = \text{dis}(\mathcal{M}, \mathcal{M}_{t-1}) + \eta \ell(\mathcal{M}, (\sqrt{\alpha_t}x_t, \sqrt{\alpha_t}\tilde{y}_t)) , \quad (2)$$

where we replaced x_t and y_t in (1) with $\sqrt{\alpha_t}x_t$ and $\sqrt{\alpha_t}\tilde{y}_t$ respectively.

Our algorithm sets the weight α_t to be the solution of the following problem,

$$\alpha_t = \arg \min_{\alpha} \mathbb{E} \left[\min_{\mathcal{M}} \tilde{C}(\mathcal{M}; \alpha) \right] ,$$

where the expectation is with respect to the labels noise distribution. That is, α_t is chosen to minimize the average (over noise) optimal (over models, per noise instance) objective over all instances of noise. In other words, we seek for the single (or shared) value α_t that performs well on average.

Let us now develop the above equation. We start with some notation, let

$$\mathcal{M}_t(\alpha) = \arg \min_{\mathcal{M}} \tilde{C}(\mathcal{M}; \alpha)$$

to be the updated model at round t . We substitute \mathcal{M}_t in (1),

$$C(\mathcal{M}_t(\alpha); \alpha) = d(\mathcal{M}_t(\alpha), \mathcal{M}_{t-1}) + \eta \ell(\mathcal{M}_t(\alpha), (x_t, y_t)).$$

Next we optimize for α_t ,

$$\alpha_t = \arg \min_{\alpha} \mathbb{E}[C(\mathcal{M}_t; \alpha)] ,$$

where the expectation is with respect to the labels noise distribution.

In this work we adapt the additive update rule from [13] to the noisy setting. The algorithm maintains a weight vector $w \in \mathbb{R}^d$. Given a new example (x_t, y_t) the algorithm predicts $\hat{y}_t = w_{t-1}^\top x_t$ and then sets the weight vector to be the minimizer of the following objective,

$$C(w) = \frac{1}{2} \|w - w_{t-1}\|^2 + \frac{1}{2r} (y_t - w^\top x_t)^2 , \quad (3)$$

where $r > 0$ is a tradeoff parameter. Note that for the special case $r \rightarrow 0$ we have that (3) becomes $\min_w \|w - w_{t-1}\|^2$ s.t. $w^\top x_t = y_t$, which recovers the normalized least mean squares (NLMS) algorithm [2]. Substituting $\sqrt{\alpha_t}x_t$ and $\sqrt{\alpha_t}\tilde{y}_t$ instead of x_t and y_t in (3) we get

$$\tilde{C}(w; \alpha_t) = \frac{1}{2} \|w - w_{t-1}\|^2 + \frac{\alpha_t}{2r} (\tilde{y}_t - w^\top x_t)^2 .$$

By setting $\nabla_w \tilde{C}(w; \alpha_t) = 0$ the update rule for w becomes

$$\begin{aligned} w_t &= w_{t-1} + \frac{(\sqrt{\alpha_t}\tilde{y}_t - w_{t-1}^\top \sqrt{\alpha_t}x_t) \sqrt{\alpha_t}x_t}{r + \|\sqrt{\alpha_t}x_t\|^2} \\ &= w_{t-1} + \frac{(\tilde{y}_t - w_{t-1}^\top x_t) x_t}{r_t + \|x_t\|^2} , \end{aligned} \quad (4)$$

where $r_t \doteq \frac{r}{\alpha_t}$. Next, we substitute (4) in (3),

$$\begin{aligned} C(w_t) &= \frac{1}{2} \left\| \frac{(\tilde{y}_t - w_{t-1}^\top x_t) x_t}{r_t + \|x_t\|^2} \right\|^2 + \frac{1}{2r} \left(y_t - w_{t-1}^\top x_t - \frac{(\tilde{y}_t - w_{t-1}^\top x_t) \|x_t\|^2}{r_t + \|x_t\|^2} \right)^2 \\ &= \frac{1}{2r} \frac{(\tilde{y}_t - w_{t-1}^\top x_t)^2 \|x_t\|^2 (r + \|x_t\|^2)}{(r_t + \|x_t\|^2)^2} + \frac{1}{2r} (y_t - w_{t-1}^\top x_t)^2 \\ &\quad - \frac{1}{r} \frac{(y_t - w_{t-1}^\top x_t) (\tilde{y}_t - w_{t-1}^\top x_t) \|x_t\|^2}{r_t + \|x_t\|^2} . \end{aligned}$$

Taking expectation and using $\mathbb{E}[\tilde{y}_t] = y_t$ and $\mathbb{E}[(\tilde{y}_t - y_t)^2] = v_t$, we get

$$\begin{aligned}\mathbb{E}[C(w_t)] &= \frac{1}{2r} \frac{\left((y_t - w_{t-1}^\top x_t)^2 + v_t\right) \|x_t\|^2 (r + \|x_t\|^2)}{\left(r_t + \|x_t\|^2\right)^2} \\ &\quad + \frac{1}{2r} (y_t - w_{t-1}^\top x_t)^2 - \frac{1}{r} \frac{(y_t - w_{t-1}^\top x_t)^2 \|x_t\|^2}{r_t + \|x_t\|^2}.\end{aligned}$$

From $\frac{d}{dr_t} (\mathbb{E}[C(w_t)]) = 0$ we get the optimal value of r_t ,

$$r_t = r + \frac{r + \|x_t\|^2}{(y_t - w_{t-1}^\top x_t)^2} v_t,$$

and

$$\alpha_t = \frac{r}{r_t} = \frac{1}{1 + \frac{r + \|x_t\|^2}{r(y_t - w_{t-1}^\top x_t)^2} v_t}. \quad (5)$$

By substituting the optimal α_t back in (4) we have that

$$w_t = w_{t-1} + \left(\frac{1}{1 + \frac{v_t}{(y_t - w_{t-1}^\top x_t)^2}} \right) \frac{(\tilde{y}_t - w_{t-1}^\top x_t) x_t}{r + \|x_t\|^2}.$$

We observe from the above update-rule that the learning rate is adapted according to the ratio of the noise variance and the algorithm's loss (with the clean label) on the current example. When the noise variance is large, compared to the loss suffered by the algorithm (with the clean label), the learning rate is small, as the uncertainty (due to label noise) of the update is large compared to the need to update (large loss).

However, the value of α_t in (5) depends on the clean label y_t which is unknown. Instead, in the experiments below we consider a few variants to replace y_t in (5). We also consider a more general form, which tries to imitate the dependence in (5) of α_t on v_t ,

$$\alpha_t = \frac{1}{1 + \beta v_t}, \quad (6)$$

for a parameter $\beta \geq 0$. The parameter β is tuned on some grid (see below). A summary of the algorithm, which we call ORS (Online Regression with Scaling), appears in Figure 1.

Remark 1 *An alternative way to derive the formula (5) is the following. Consider minimization of $C(w)$ in (3), which yields the update rule*

$$w_t = w_{t-1} + \frac{(y_t - w_{t-1}^\top x_t) x_t}{r + \|x_t\|^2}.$$

Parameter: $r > 0, \beta \geq 0$

Initialize: Set $w_0 = 0 \in \mathbb{R}^d$

For $t = 1, \dots, T$ **do**

- Receive an instance $x_t \in \mathbb{R}^d$
- Output prediction $\hat{y}_t = w_{t-1}^\top x_t$
- Receive a noisy label $\tilde{y}_t \in \mathbb{R}$
- Set scaling factor α_t according to (5) or (6)
- Update

$$w_t \leftarrow w_{t-1} + \frac{(\tilde{y}_t - w_{t-1}^\top x_t) x_t}{\frac{r}{\alpha_t} + \|x_t\|^2}$$

Fig. 1. ORS - Online Regression with Scaling.

Substitute $\sqrt{\alpha_t}x_t$ and $\sqrt{\alpha_t}\tilde{y}_t$ instead of x_t and y_t (and using \tilde{w}_t to denote the new vector based on the noisy label) we get,

$$\tilde{w}_t = w_{t-1} + \frac{(\tilde{y}_t - w_{t-1}^\top x_t) x_t}{\frac{r}{\alpha_t} + \|x_t\|^2}.$$

We minimize the mean-square-error,

$$\begin{aligned} \alpha_t &= \arg \min_{\alpha} \mathbb{E} \|\tilde{w}_t - w_t\|^2 \\ &= \arg \min_{\alpha} \mathbb{E} \left[\left(\frac{\tilde{y}_t - w_{t-1}^\top x_t}{\frac{r}{\alpha} + \|x_t\|^2} - \frac{y_t - w_{t-1}^\top x_t}{r + \|x_t\|^2} \right)^2 \|x_t\|^2 \right] \\ &= \arg \min_{\alpha} \mathbb{E} \left[\left(\frac{\tilde{y}_t - w_{t-1}^\top x_t}{\frac{r}{\alpha} + \|x_t\|^2} \right)^2 - 2 \frac{(\tilde{y}_t - w_{t-1}^\top x_t)(y_t - w_{t-1}^\top x_t)}{\left(\frac{r}{\alpha} + \|x_t\|^2 \right) (r + \|x_t\|^2)} \right. \\ &\quad \left. + \left(\frac{y_t - w_{t-1}^\top x_t}{r + \|x_t\|^2} \right)^2 \right] \\ &= \arg \min_{\alpha} \left(\frac{(y_t - w_{t-1}^\top x_t)^2 + v_t}{\left(\frac{r}{\alpha} + \|x_t\|^2 \right)^2} - 2 \frac{(y_t - w_{t-1}^\top x_t)^2}{\left(\frac{r}{\alpha} + \|x_t\|^2 \right) (r + \|x_t\|^2)} \right). \end{aligned}$$

Setting the derivative of the objective to 0 we get

$$\begin{aligned} &\frac{2r}{\alpha^2} \frac{(y_t - w_{t-1}^\top x_t)^2 + v_t}{\left(\frac{r}{\alpha} + \|x_t\|^2 \right)^3} - \frac{2r}{\alpha^2} \frac{(y_t - w_{t-1}^\top x_t)^2}{\left(\frac{r}{\alpha} + \|x_t\|^2 \right)^2 (r + \|x_t\|^2)} = 0 \\ \Rightarrow &\frac{(y_t - w_{t-1}^\top x_t)^2 + v_t}{\frac{r}{\alpha} + \|x_t\|^2} = \frac{(y_t - w_{t-1}^\top x_t)^2}{r + \|x_t\|^2}. \end{aligned}$$

Solving for α we get (5).

3.1 Discussion

There is a line of research in adaptive filtering theory that discusses NLMS algorithms with *variable* step size (VSS) [12, 7, 17, 25, 22, 18]. In all these works the update rule is *assumed* beforehand to be of the additive form $w_t = w_{t-1} + \mu_t x_t e_t$ where $e_t = \tilde{y}_t - w_{t-1}^\top x_t$ is the error signal, and the only question is how to choose the step size to control the tradeoff between fast convergence rate and low steady-state misalignment. A common approach is to require $y_t = w_t^\top x_t$, which leads to $\mu_t = \left(1 - \frac{\varepsilon_t}{e_t}\right) / \|x_t\|^2$ where $\varepsilon_t = \tilde{y}_t - y_t$ is the instantaneous noise. As this result is impractical from many points of view (ε_t is unknown, μ_t could become negative), a few modifications were advised. One approach [12] suggested to employ a bound on the instantaneous noise, and also limit μ_t below by 0. Another approach [1], is to use $\mu_t = \left(1 - \frac{\sqrt{v_t}}{e_t}\right) / \|x_t\|^2$ where it is assumed that the noise variance v_t is known. In case the noise variance is unknown some heuristics have been proposed how to estimate it in some specific adaptive filtering applications (e.g. in echo cancellation, it can be estimated during silences [18]). Valin and Collins [21] proposed another modification with the INLMS algorithm, which adapts the learning rate to achieve fast convergence.

In contrary to this previous work we do not assume any type of the update rule. We solve an optimization problem, which can also be generalized to other loss functions and dissimilarity measures. For the case of square loss and euclidean distance we derive an update rule which turns out to be NLMS-like with step size $\mu_t = 1 / \left(\frac{r}{\alpha_t} + \|x_t\|^2\right)$. Here the regularization parameter r is modulated by the label noise variance according to (5).

4 Experimental Study

We evaluated the update rule (4) with adaptive scaling (5) and (6). In Section 4.1 we consider the case of a single noisy label, and compare a few variants in case the noise variance is known or unknown. Then, in Section 4.2 we assume that $k > 1$ noisy labels per example can be sampled, and propose a label sampling rule that uses the instantaneous margin, which together with scaling outperforms other baselines.

4.1 Single feedback label

We use two synthetic and one real-world datasets, and compare the mean square error during the algorithm’s run-time. We distinguish between two settings:

Known variance: in this case we assume that the label noise variance is known, so that we can employ it in (5). We consider the following variants:

- ‘Clean feedback’ - there is no label noise. This is the *only* variant without label noise and is used for reference.
- ‘Noisy feedback’ - the algorithm does not apply scaling for examples ($\alpha_t = 1$).

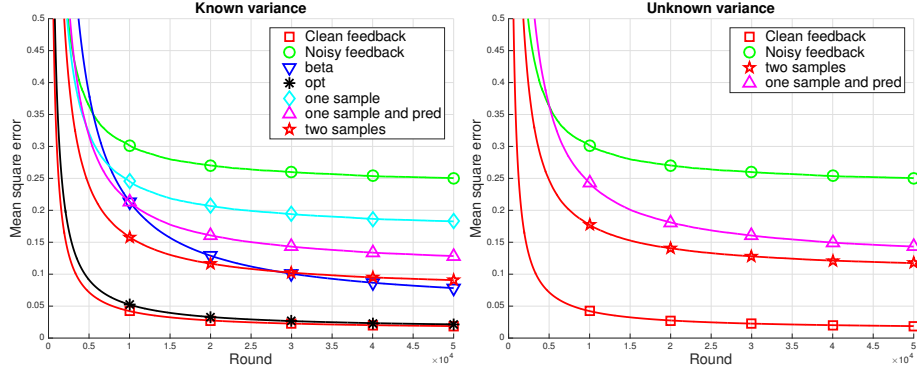


Fig. 2. Synthetic dataset. Left - known variance, right - unknown variance

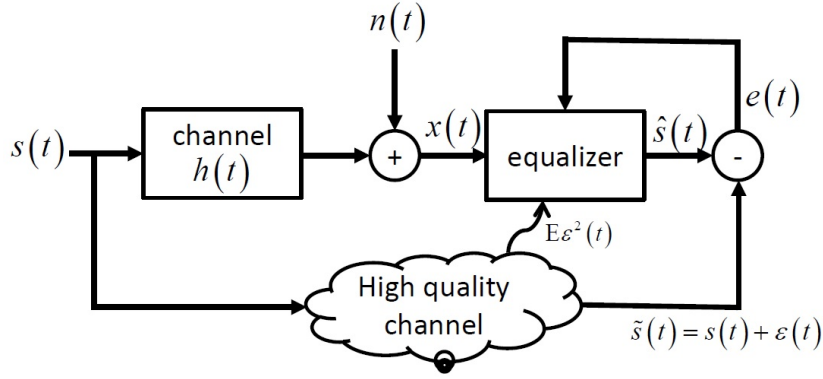


Fig. 3. Communication scheme with noisy equalizer feedback.

- ‘beta’ - (6) is used as a scaling rule.
- ‘opt’ - (5) is used as a scaling rule, when we assume that y_t is *known*. Note that the update (4) is still performed with a noisy label \tilde{y}_t .
- ‘one sample’ - (5) is used, approximating y_t with \tilde{y}_t .
- ‘one sample and pred’ - (5) is used, approximating y_t with the average of the label \tilde{y}_t and the prediction \hat{y}_t .
- ‘two samples’ - in this variant we assume that the algorithm has access to 2 samples of the label. In this case we employ the average of the labels as an approximation to y_t in (5) and \tilde{y}_t in (4). The motivation for this variant will become clear when we discuss the unknown variance algorithms now.

Unknown variance: in this case we assume that the label’s noise variance is unknown, and to employ (5) the algorithm should estimate the variance in some way. We consider the following variants:

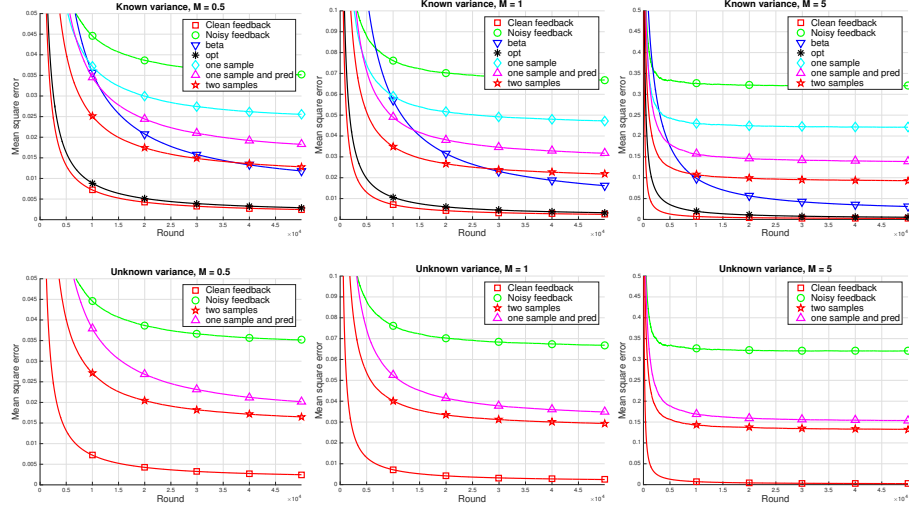


Fig. 4. Communication scheme evaluation. Top - known variance for different noise levels, bottom - unknown variance for different noise levels

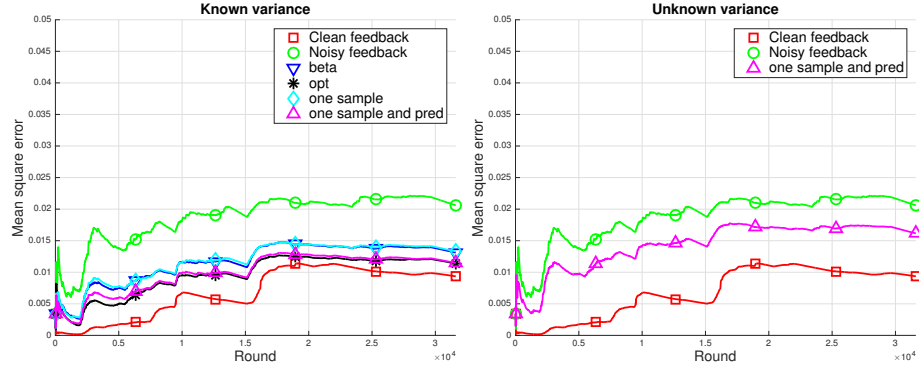


Fig. 5. Adaptive speech filter evaluation with known variance (left) and unknown variance (right).

- 'Clean feedback' - there is no label noise. This is the *only* variant without label noise and used for reference.
- 'Noisy feedback' - the algorithm does not apply scaling of examples ($\alpha_t = 1$).
- 'one sample and pred' - (5) is used, when we employ \tilde{y}_t and the prediction \hat{y}_t to approximate $y_t = (\tilde{y}_t + \hat{y}_t)/2$ and $v_t = (\tilde{y}_t - \hat{y}_t)^2/4$.
- 'two samples' - in this variant we assume that the algorithm has access to 2 noisy samples of the label - $\tilde{y}_{t,1}$ and $\tilde{y}_{t,2}$. We approximate $y_t = (\tilde{y}_{t,1} + \tilde{y}_{t,2})/2$ and $v_t = (\tilde{y}_{t,1} - \tilde{y}_{t,2})^2/4$ in (5). Also we employ $(\tilde{y}_{t,1} + \tilde{y}_{t,2})/2$ in (4).

In our first experiment we used a synthetic dataset with 50,000 examples of dimension $d = 20$. The inputs $x_t \in \mathbb{R}^{20}$ were drawn from a zero-mean unit-covariance Gaussian distribution. The target $u \in \mathbb{R}^{20}$ is a zero-mean unit-covariance Gaussian vector. The labels were set according to $y_t = x_t^\top u + n_t$ where $n_t \sim \mathcal{N}(0, 0.01)$. We introduced label noise $\tilde{y}_t = y_t + \epsilon_t$ where $v_t = \mathbb{E}\epsilon_t^2$ was chosen from uniform distribution on $[0, M]$ for $M = 5$. The parameters r, β were tuned using a single random sequence. The experiment was repeated 20 times and we report the average mean-square-error (error bars are very small and thus do not appear in the plots).

The results are summarized in Figure 2. For the known variance case we observe that (6) outperforms other variants. However, it requires tuning of the β parameter. In addition, taking into account the algorithm’s prediction can help in the long run (‘one sample and pred’ curve). For the unknown variance case it is clearly better to employ (5) with the prediction (‘one sample and pred’ curve) than doing nothing. In the long run it is not far from the case when two samples are available.

In our second experiment we considered the communication scheme in Figure 3. The 50,000 input signal samples $s(t)$ were drawn from a zero-mean unit-variance Gaussian distribution. The channel $h(t)$ is a typical band-limited channel $h(t) = \text{sinc}(t)$ for $t = 0, 0.5, 1, \dots, 4.5$ (dimension 10). We set $n(t) \sim \mathcal{N}(0, 4 \times 10^{-4})$. The equalizer performs deconvolution so that its output should be a good estimate of the transmitted signal $s(t)$. To this end, the equalizer uses $d = 20$ consecutive samples of $x(t)$ for each output sample. The equalizer’s feedback is noisy, $\tilde{s}_t = s_t + \epsilon_t$ where $v_t = \mathbb{E}\epsilon_t^2$ was chosen from uniform distribution on $[0, M]$. We considered a few values of M (0.5, 1, 5). The parameters r, β were tuned using a single random sequence. The experiment was repeated 20 times and we report the average mean-square-error (error bars are very small and thus do not appear in the plots).

The results for different values of M are summarized in Figure 4, note the difference in scale (y-axis) for the three values of M , i.e. across each line. Clearly, for more label noise (i.e. larger M) the improvement with scaling compared to without scaling is bigger.

In our third experiment we considered training of adaptive speech filter. The general scheme is similar to Figure 3. A clean speech signal was passed through a typical room impulse response $h(t)$ of length 1024. We set $n(t) \sim \mathcal{N}(0, 4 \times 10^{-4})$. The purpose is to train adaptive filter that will recover the original speech signal. To this end, the filter uses $d = 1024$ consecutive samples of $x(t)$ for each output sample. We assume that the feedback of the filter was recorded with hammer noise in the background. The parameters r, β were tuned on 20% of the signal. The results are summarized in Figure 5. Again, scaling can improve the performance of the adaptive filter. We emphasize, this approach does not have any additional information beyond the noisy label, and yet it works much better than just using the noisy label.

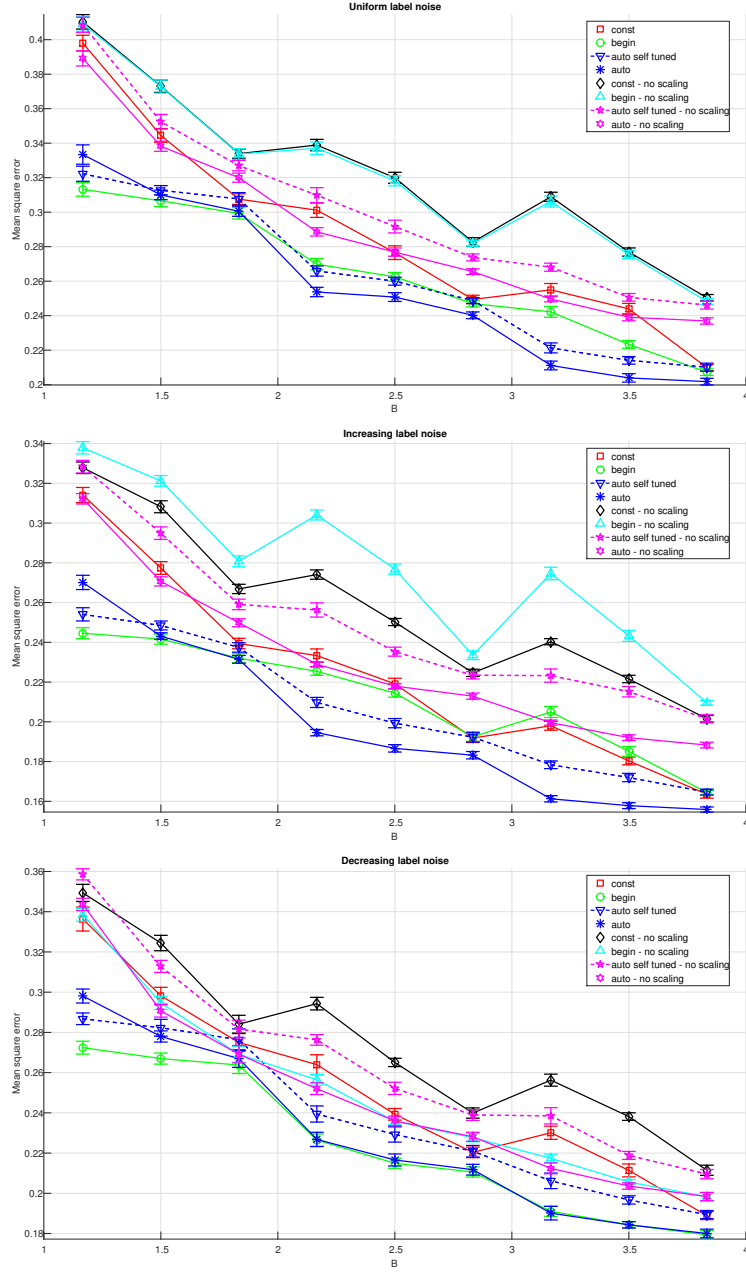


Fig. 6. Synthetic dataset with variable labels budget.

4.2 From 1 to k labels

In this part we assume that the algorithm has access to a single feedback label or more each round. Specifically, there is a budget parameter B which defines

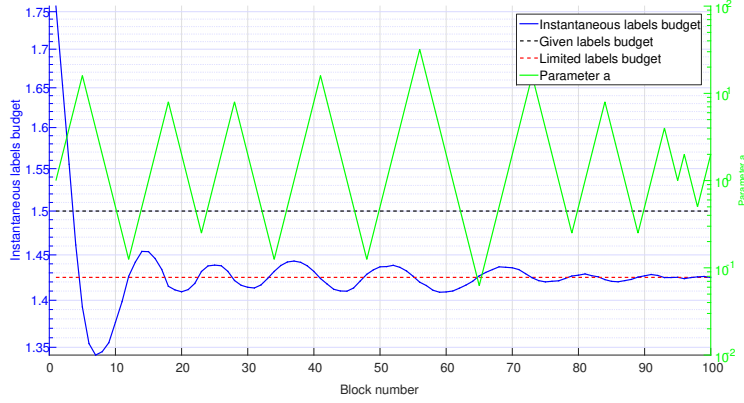


Fig. 7. Evaluation of the self-tuned scheme.

the average number of labels per round. In other words, for n rounds the total number of labels is $B \times n$. We assume that each round the algorithm has access to a single noisy label or $k = \lceil B \rceil$ noisy labels. Note that for $k > 1$ the noise variance is reduced by a factor of k , which allows a more accurate update.

We employ the first synthetic dataset from Section 4.1, with $n = 10000$ examples. In addition to the uniform noise variance distribution (as described in Section 4.1) we consider also the cases of linearly increasing noise variance (from 0 to 5) and linearly decreasing noise variance (from 5 to 0).

We consider the following strategies for spending the given labels budget B :

- ‘const’ - in this case, in each round, k labels are sampled with probability $p = (B - 1)/(k - 1)$.
- ‘begin’ - in this case, k labels are sampled at the beginning of the algorithm’s runtime. For total n rounds, k labels are sampled for the first $n \times p$ rounds.
- ‘auto’ - in this case, in each round 1 label \tilde{y}_t is sampled and then we use the prediction \hat{y}_t to decide whether to sample $k - 1$ more labels. Our decision rule is probabilistic, and we decide to stay with 1 label with probability $\frac{a}{a + (\hat{y}_t - \tilde{y}_t)^2}$ for a positive parameter a . The parameter a is tuned manually so that the total number of labels is (close to) $B \times n$. A disadvantage of this strategy is that given some budget B it is not clear how to set a to spend the budget. A similar issue has been reported in randomized selective sampling algorithms [3]. To solve it, we propose the next strategy.
- ‘auto self tuned’ - this strategy is similar to ‘auto’, yet the algorithm receives as input the given budget B and tunes on the fly the parameter a as following: we start by setting $a = 1$, then each block of 100 rounds we check the instantaneous labels budget until the current round. If it is larger than the given budget, we try to decrease it by setting $a \leftarrow 2a$. However, if the instantaneous budget is smaller than the given budget we try to increase it by setting $a \leftarrow a/2$. In order to make sure that we do not sample more labels than the given budget, we take a safety margin and actually guide

the algorithm to sample 5% less than the given budget. We call it limited budget, which equals $0.95B$.

We run the update (4) with scaling (5). For rounds where we have only a single label, we employ the prediction and update as described above for the 'one sample and pred' case for unknown variance. For rounds where we have k labels, we employ the sample mean (in (4) and (5)) and the sample variance (in (5)). In addition, we run a version of the algorithm (4) without scaling (i.e. $\alpha_t = 1$), which corresponds to "no scaling" lines in the plots. In this case the update (4) is performed with the single label or the mean for k labels.

The results are reported in Figure 6, where the error bars correspond to the 95% confidence interval over 20 runs. Clearly, for all 3 noise models the scaling approach outperforms the 'no scaling' approach. Also, the 'auto' sampling strategy usually outperform 'begin' and 'const'. We note also that the self tuned version ('auto self tuned'), which does not require any manual tuning and samples 5% less labels on average, works quite well and not too much worse than 'auto'.

For decreasing noise it is beneficial to sample k labels at the beginning. This is expected as it allows learning with effectively smaller noise at the beginning, bringing the algorithm closer (compared to using only a single label) to a good model. Then the algorithm can employ the predictions together with only a single noisy label.

From the uniform noise plot, we observe that for $B > 2$ 'auto' outperform 'begin' (with scaling). This is expected, as for larger B (i.e. more labels) the algorithm makes more accurate predictions, which in turn leads to a better estimation of the margin in the 'auto' sampling rule. Also note that for the 'no scaling' case, 'begin' performs like 'const'. This is because the 'no scaling' algorithm does not use the predictions to improve its performance. From the increasing noise plot, we observe that the 'begin' strategy, which spends the labels budget at the beginning, does not perform well. This is expected as we need more labels where the noise variance is large.

Finally, in Figure 7 we demonstrate how the instantaneous budget and the parameter a change during a run of the 'auto self tuned' version (for a given budget $B = 1.5$). We note that the instantaneous budget converges close to the limited budget, and any case we do not sample more labels than the given budget.

5 Summary and Future Directions

We proposed a shrinkage scheme for online regression (adaptive filtering) with noisy feedback. Our algorithms are theoretically motivated by minimizing the average (over noise) optimal objective which captures the change of a model and loss. Our algorithms adapt the learning rate by taking in account the tradeoff between the label noise and the need to update using this label. We considered both cases, when the noise-variance per instance is known, or when it is estimated. We showed empirically the usefulness of our approach on synthetic and

real speech data, even when the noise variance is unknown. We also considered the case when an algorithm is allowed to sample a few noisy labels per example, yet with limited total budget for all examples. We proposed a simple and effective sampling rule to decide how to spend the budget. This sampling rule, together with the shrinkage scheme, outperforms other approaches.

There are many directions for future research. First, analyzing our approach, e.g. in the regret bound model. Second, considering other types of noise, rather than additive. Third, developing efficient label sampling rules that can sample any number of noisy labels (not only 1 or k) given some budget. Finally, evaluating theoretically and empirically the robustness of our approach to the accurate estimation of the noise variance (i.e. the variance is known with some error).

References

1. Benesty, J., Rey, H., Vega, L.R., Tressens, S.: A nonparametric VSS NLMS algorithm. *IEEE Signal Process. Lett.* 13(10), 581–584 (2006)
2. Bershadt, N.J.: Analysis of the normalized lms algorithm with gaussian inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34(4), 793–806 (1986)
3. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Worst-case analysis of selective sampling for linear classification. *Journal of Machine Learning Research* 7, 1205–1230 (2006)
4. Cesa-Bianchi, N., Long, P.M., Warmuth, M.K.: Worst case quadratic loss bounds for on-line prediction of linear functions by gradient descent. Technical Report IR-418, University of California, Santa Cruz, CA, USA (1993)
5. Cesa-Bianchi, N., Shalev-Shwartz, S., Shamir, O.: Online learning of noisy data. *IEEE Transactions on Information Theory* 57(12), 7907–7931 (2011)
6. Chawla, S., Dwork, C., McSherry, F., Smith, A.D., Wee, H.: Toward privacy in public databases. In: *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*. pp. 363–385 (2005)
7. Ciochină, S., Paleologu, C., Benesty, J.: An optimized nlms algorithm for system identification. *Signal Process.* 118(C), 115–121 (Jan 2016)
8. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. *JMLR* 7, 551–585 (2006)
9. Crammer, K., Kulesza, A., Dredze, M.: Adaptive regularization of weighted vectors. In: *Advances in Neural Information Processing Systems* 23 (2009)
10. Dredze, M., Crammer, K., Pereira, F.: Confidence-weighted linear classification. In: *International Conference on Machine Learning* (2008)
11. Forster, J.: On relative loss bounds in generalized linear regression. In: *Fundamentals of Computation Theory (FCT)* (1999)
12. Gollamudi, S., Nagaraj, S., Kapoor, S., Huang, Y.F.: Set-membership filtering and a set-membership normalized lms algorithm with an adaptive step size. *IEEE Signal Processing Letters* 5(5), 111–114 (1998)
13. Kivinen, J., K.Warmuth, M.: Exponential gradient versus gradient descent for linear predictors. *Information and Computation* 132, 132–163 (1997)
14. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.* 38(3), 447–461 (2016)

15. Moroshko, E., Crammer, K.: Weighted last-step min-max algorithm with improved sub-logarithmic regret. *Theor. Comput. Sci.* 558, 107–124 (2014)
16. Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A.: Learning with noisy labels. In: *Advances in Neural Information Processing Systems* 26, pp. 1196–1204 (2013)
17. Paleologu, C., Ciochină, S., Benesty, J., Grant, S.L.: An overview on optimized NLMS algorithms for acoustic echo cancellation. *EURASIP Journal on Applied Signal Processing* 2015, 97 (Dec 2015)
18. Paleologu, C., Ciochina, S., Benesty, J.: Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation. *IEEE Signal Process. Lett.* 15, 5–8 (2008)
19. Sayed, A.H.: *Adaptive Filters*. John Wiley & Sons, Inc. (2008)
20. Vaits, N., Crammer, K.: Re-adapting the regularization of weights for non-stationary regression. In: *The 22nd International Conference on Algorithmic Learning Theory*. ALT '11 (2011)
21. Valin, J., Collings, I.B.: Interference-normalized least mean square algorithm. *IEEE Signal Process. Lett.* 14(12), 988–991 (2007)
22. Vega, L., Rey, H., Benesty, J., Tressens, S.: A new robust variable step-size nlms algorithm. *Trans. Sig. Proc.* 56(5), 1878–1893 (May 2008)
23. Vovk, V.: Competitive on-line statistics. *International Statistical Review* 69 (2001)
24. Widrow, B., M.E. Hoff, J.: *Adaptive switching circuits* (1960)
25. Yu, Y., Zhao, H.: A novel variable step size NLMS algorithm based on the power estimate of the system noise. *CoRR* abs/1504.05323 (2015)