Learning with Annotation Noise

Eyal Beigman
Olin Business School
Washington University in St. Louis
beigman@wustl.edu

Beata Beigman Klebanov
Kellogg School of Management
Northwestern University
beata@northwestern.edu

Abstract

It is usually assumed that the kind of noise existing in annotated data is random classification noise. Yet there is evidence that differences between annotators are not always random attention slips but could result from different biases towards the classification categories, at least for the harder-to-decide cases. Under an annotation generation model that takes this into account, there is a hazard that some of the training instances are actually hard cases with unreliable annotations. We show that these are relatively unproblematic for an algorithm operating under the 0-1 loss model, whereas for the commonly used voted perceptron algorithm, hard training cases could result in incorrect prediction on the *uncontroversial* cases at test time.

1 Introduction

It is assumed, often tacitly, that the kind of noise existing in human-annotated datasets used in computational linguistics is random classification noise (Kearns, 1993; Angluin and Laird, 1988), resulting from annotator attention slips randomly distributed across instances. For example, Osborne (2002) evaluates noise tolerance of shallow parsers, with random classification noise taken to be "crudely approximating annotation errors." It has been shown, both theoretically and empirically, that this type of noise is tolerated well by the commonly used machine learning algorithms (Cohen, 1997; Blum et al., 1996; Osborne, 2002; Reidsma and Carletta, 2008).

Yet this might be overly optimistic. Reidsma and op den Akker (2008) show that apparent differences between annotators are not random slips of attention but rather result from different biases annotators might have towards the classification categories. When training data comes from one annotator and test data from another, the first annotator's biases are sometimes systematic enough for a machine learner to pick them up, with detrimental results for the algorithm's performance on the test data. A small subset of doubly annotated data (for inter-annotator agreement check) and large chunks of singly annotated data (for training algorithms) is not uncommon in computational linguistics datasets; such a setup is prone to problems if annotators are differently biased.[1]

Annotator bias is consistent with a number of noise models. For example, it could be that an annotator's bias is exercised on each and every instance, making his preferred category likelier for any instance than in another person's annotations. Another possibility, recently explored by Beigman Klebanov and Beigman (2009), is that some items are really quite clear-cut for an annotator with any bias, belonging squarely within one particular category. However, some instances – termed **hard cases** therein – are harder to decide upon, and this is where various preferences and biases come into play. In a metaphor annotation study reported by Beigman Klebanov et al. (2008), certain markups received overwhelming annotator support when people were asked to validate annotations after a certain time delay. Other instances saw opinions split; moreover, Beigman Klebanov et al. (2008) observed cases where people retracted their own earlier annotations.

To start accounting for such annotator behavior, Beigman Klebanov and Beigman (2009) proposed a model where instances are either **easy**, and then all annotators agree on them, or **hard**, and then each annotator flips his or her own coin to de-

[1] The different biases might not amount to much in the small doubly annotated subset, resulting in acceptable inter-annotator agreement; yet when enacted throughout a large number of instances they can be detrimental from a machine learner's perspective.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 280–287,
Suntec, Singapore, 2-7 August 2009.

cide on a label (each annotator can have a different "coin" reflecting his or her biases). For annotations generated under such a model, there is a danger of hard instances posing as easy – an observed agreement between annotators being a result of all coins coming up heads by chance. They therefore define the expected proportion of hard instances in agreed items as **annotation noise**. They provide an example from the literature where an annotation noise rate of about 15% is likely.

The question addressed in this article is: How problematic is learning from training data with annotation noise? Specifically, we are interested in estimating the degree to which performance on easy instances at test time can be hurt by the presence of hard instances in training data.

Definition 1 *The* **hard case bias***, τ, is the portion of easy instances in the test data that are misclassified as a result of hard instances in the training data.*

This article proceeds as follows. First, we show that a machine learner operating under a 0-1 loss minimization principle could sustain a hard case bias of $\theta(\frac{1}{\sqrt{N}})$ in the worst case. Thus, while annotation noise is hazardous for small datasets, it is better tolerated in larger ones. However, 0-1 loss minimization is computationally intractable for large datasets (Feldman et al., 2006; Guruswami and Raghavendra, 2006); substitute loss functions are often used in practice. While their tolerance to random classification noise is as good as for 0-1 loss, their tolerance to annotation noise is worse. For example, the perceptron family of algorithms handle random classification noise well (Cohen, 1997). We show in section 3.4 that the widely used Freund and Schapire (1999) voted perceptron algorithm could face a constant hard case bias when confronted with annotation noise in training data, irrespective of the size of the dataset. Finally, we discuss the implications of our findings for the practice of annotation studies and for data utilization in machine learning.

2 0-1 Loss

Let a sample be a sequence $x_1, \ldots, x_N$ drawn uniformly from the d-dimensional discrete cube $I_d = \{-1, 1\}^d$ with corresponding labels $y_1, \ldots, y_N \in \{-1, 1\}$. Suppose further that the learning algorithm operates by finding a hyperplane (w, ψ), $w \in \mathbb{R}^d, \psi \in \mathbb{R}$, that minimizes the empirical error $L(w, \psi) = \sum_{j=1\ldots N}[y_j - \text{sgn}(\sum_{i=1\ldots d} x_j^i w^i - \psi)]^2$. Let there be H hard cases, such that the annotation noise is $\gamma = \frac{H}{N}$.[2]

Theorem 1 *In the worst case configuration of instances a hard case bias of $\tau = \theta(\frac{1}{\sqrt{N}})$ cannot be ruled out with constant confidence.*

Idea of the proof: We prove by explicit construction of an adversarial case. Suppose there is a plane that perfectly separates the easy instances. The $\theta(N)$ hard instances will be concentrated in a band parallel to the separating plane, that is near enough to the plane so as to trap only about $\theta(\sqrt{N})$ easy instances between the plane and the band (see figure 1 for an illustration). For a random labeling of the hard instances, the central limit theorem shows there is positive probability that there would be an imbalance between $+1$ and -1 labels in favor of -1s on the scale of $\sqrt{N}$, which, with appropriate constants, would lead to the movement of the empirically minimal separation plane to the right of the hard case band, misclassifying the trapped easy cases.

Proof: Let $v = v(x) = \sum_{i=1\ldots d} x^i$ denote the sum of the coordinates of an instance in I_d and take $\lambda_e = \sqrt{d} \cdot F^{-1}(\sqrt{\gamma} \cdot 2^{-\frac{d}{2}} + \frac{1}{2})$ and $\lambda_h = \sqrt{d} \cdot F^{-1}(\gamma + \sqrt{\gamma} \cdot 2^{-\frac{d}{2}} + \frac{1}{2})$, where $F(t)$ is the cumulative distribution function of the normal distribution. Suppose further that instances x_j such that $\lambda_e < v_j < \lambda_h$ are all and only hard instances; their labels are coinflips. All other instances are easy, and labeled $y = y(x) = \text{sgn}(v)$. In this case, the hyperplane $\frac{1}{\sqrt{d}}(1 \ldots 1)$ is the true separation plane for the easy instances, with $\psi = 0$. Figure 1 shows this configuration.

According to the central limit theorem, for d, N large, the distribution of v is well approximated by $\mathcal{N}(0, \sqrt{d})$. If $N = c_1 \cdot 2^d$, for some $0 < c_1 < 4$, the second application of the central limit theorem ensures that, with high probability, about $\gamma N = c_1 \gamma 2^d$ items would fall between λ_e and λ_h (all hard), and $\sqrt{\gamma} \cdot 2^{-\frac{d}{2}} N = c_1 \sqrt{\gamma 2^d}$ would fall between 0 and λ_e (all easy, all labeled $+1$).

Let Z be the sum of labels of the hard cases, $Z = \sum_{i=1\ldots H} y_i$. Applying the central limit theorem a third time, for large N, Z will, with a high probability, be distributed approximately as

[2] In Beigman Klebanov and Beigman (2009), annotation noise is defined as percentage of hard instances in the agreed annotations; this implies noise measurement on multiply annotated material. When there is just one annotator, no distinction between easy vs hard instances can be made; in this sense, all hard instances are posing as easy.

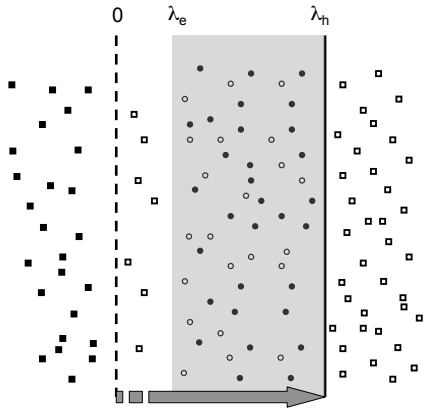

Figure 1: The adversarial case for 0-1 loss. Squares correspond to easy instances, circles – to hard ones. Filled squares and circles are labeled -1, empty ones are labeled $+1$.

$\mathcal{N}(0, \sqrt{\gamma N})$. This implies that a value as low as -2σ cannot be ruled out with high (say 95%) confidence. Thus, an imbalance of up to $2\sqrt{\gamma N}$, or of $2\sqrt{c_1\gamma 2^d}$, in favor of -1s is possible.

There are between 0 and λ_h about $2\sqrt{c}_1\sqrt{\gamma 2^d}$ more -1 hard instances than $+1$ hard instances, as opposed to $c_1\sqrt{\gamma 2^d}$ easy instances that are all $+1$. As long as $c_1 < 2\sqrt{c}_1$, i.e. $c_1 < 4$, the empirically minimal threshold would move to λ_h, resulting in a hard case bias of $\tau = \frac{\sqrt{\gamma}\sqrt{c_1 2^d}}{(1-\gamma)\cdot c_1 2^d} = \theta(\frac{1}{\sqrt{N}})$.

To see that this is the worst case scenario, we note that 0-1 loss sustained on $\theta(N)$ hard cases is the order of magnitude of the possible imbalance between -1 and $+1$ random labels, which is $\theta(\sqrt{N})$. For hard case loss to outweigh the loss on the misclassified easy instances, there cannot be more than $\theta(\sqrt{N})$ of the latter $\square$

Note that the proof requires that $N = \theta(2^d)$ namely, that asymptotically the sample includes a fixed portion of the instances. If the sample is asymptotically smaller, then λ_e will have to be adjusted such that $\lambda_e = \sqrt{d} \cdot F^{-1}(\theta(\frac{1}{\sqrt{N}}) + \frac{1}{2})$.

According to theorem 1, for a 10K dataset with 15% hard case rate, a hard case bias of about 1% cannot be ruled out with 95% confidence.

Theorem 1 suggests that annotation noise as defined here is qualitatively different from more malicious types of noise analyzed in the agnostic learning framework (Kearns and Li, 1988; Haussler, 1992; Kearns et al., 1994), where an adversary can not only choose the placement of the hard cases, but also their labels. In worst case, the 0-1 loss model would sustain a constant rate of error due to malicious noise, whereas annotation noise is tolerated quite well in large datasets.

3 Voted Perceptron

Freund and Schapire (1999) describe the *voted perceptron*. This algorithm and its many variants are widely used in the computational linguistics community (Collins, 2002a; Collins and Duffy, 2002; Collins, 2002b; Collins and Roark, 2004; Henderson and Titov, 2005; Viola and Narasimhan, 2005; Cohen et al., 2004; Carreras et al., 2005; Shen and Joshi, 2005; Ciaramita and Johnson, 2003). In this section, we show that the voted perceptron can be vulnerable to annotation noise. The algorithm is shown below.

Algorithm 1 Voted Perceptron

Training
Input: a labeled training set $(x_1, y_1), \ldots, (x_N, y_N)$
Output: a list of perceptrons $w_1, \ldots, w_N$

Initialize: $t \leftarrow 0$; $w_1 \leftarrow 0$; $\psi_1 \leftarrow 0$
for $t = 1 \ldots N$ **do**
  $\hat{y}_t \leftarrow \text{sign}(\langle w_t, x_t\rangle + \psi_t)$
  $w_{t+1} \leftarrow w_t + \frac{y_t - \hat{y}_t}{2} \cdot x_t$
  $\psi_{t+1} \leftarrow \psi_t + \frac{y_t - \hat{y}_t}{2} \cdot \langle w_t, x_t\rangle$
end for

Forecasting
Input: a list of perceptrons $w_1, \ldots, w_N$
  an unlabeled instance x
Output: A forecasted label y

$\hat{y} \leftarrow \sum_{t=1}^{N} \text{sign}(\langle w_t, x_t\rangle + \psi_t)$
$y \leftarrow \text{sign}(\hat{y})$

The voted perceptron algorithm is a refinement of the perceptron algorithm (Rosenblatt, 1962; Minsky and Papert, 1969). Perceptron is a dynamic algorithm; starting with an initial hyperplane w_0, it passes repeatedly through the labeled sample. Whenever an instance is misclassified by w_t, the hyperplane is modified to adapt to the instance. The algorithm terminates once it has passed through the sample without making any classification mistakes. The algorithm terminates iff the sample can be separated by a hyperplane, and in this case the algorithm finds a separating hyperplane. Novikoff (1962) gives a bound on the number of iterations the algorithm goes through before termination, when the sample is separable by a margin.

The perceptron algorithm is vulnerable to noise, as even a little noise could make the sample inseparable. In this case the algorithm would cycle indefinitely never meeting termination conditions, w_t would obtain values within a certain dynamic range but would not converge. In such setting, imposing a stopping time would be equivalent to drawing a random vector from the dynamic range.

Freund and Schapire (1999) extend the perceptron to inseparable samples with their voted perceptron algorithm and give theoretical generalization bounds for its performance. The basic idea underlying the algorithm is that if the dynamic range of the perceptron is not too large then w_t would classify most instances correctly most of the time (for most values of t). Thus, for a sample $x_1, \ldots, x_N$ the new algorithm would keep track of $w_0, \ldots, w_N$, and for an unlabeled instance x it would forecast the classification most prominent amongst these hyperplanes.

The bounds given by Freund and Schapire (1999) depend on the *hinge loss* of the dataset. In section 3.2 we construct a difficult setting for this algorithm. To prove that voted perceptron would suffer from a constant hard case bias in this setting using the exact dynamics of the perceptron is beyond the scope of this article. Instead, in section 3.3 we provide a lower bound on the hinge loss for a simplified model of the perceptron algorithm dynamics, which we argue would be a good approximation to the true dynamics in the setting we constructed. For this simplified model, we show that the hinge loss is large, and the bounds in Freund and Schapire (1999) cannot rule out a constant level of error regardless of the size of the dataset. In section 3.4 we study the dynamics of the model and prove that $\tau = \theta(1)$ for the adversarial setting.

3.1 Hinge Loss

Definition 2 *The* **hinge loss** *of a labeled instance (x, y) with respect to hyperplane (w, ψ) and margin $\delta > 0$ is given by $\zeta = \zeta(\psi, \delta) = \max(0, \delta - y \cdot (\langle w, x \rangle - \psi))$.*

ζ measures the distance of an instance from being classified correctly with a δ margin. Figure 2 shows examples of hinge loss for various data points.

Theorem 2 (Freund and Schapire (1999)) *After one pass on the sample, the probability that the voted perceptron algorithm does not predict correctly the label of a test instance x_{N+1} is bounded by $\frac{2}{N+1}\mathbb{E}_{N+1}\left[\frac{d+D}{\delta}\right]^2$ where $D = D(w, \psi, \delta) = \sqrt{\sum_{i=1}^{N} \zeta_i^2}$.*

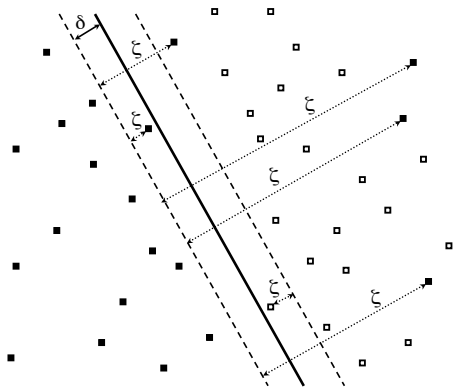

Figure 2: Hinge loss ζ for various data points incurred by the separator with margin δ.

This result is used to explain the convergence of weighted or voted perceptron algorithms (Collins, 2002a). It is useful as long as the expected value of D is not too large. We show that in an adversarial setting of the annotation noise D is large, hence these bounds are trivial.

3.2 Adversarial Annotation Noise

Let a sample be a sequence $x_1, \ldots, x_N$ drawn uniformly from I_d with $y_1, \ldots, y_N \in \{-1, 1\}$. Easy cases are labeled $y = y(x) = \text{sgn}(v)$ as before, with $v = v(x) = \sum_{i=1\ldots d} x^i$. The true separation plane for the easy instances is $w^* = \frac{1}{\sqrt{d}}(1 \ldots 1)$, $\psi^* = 0$. Suppose hard cases are those where $v(x) > c_1\sqrt{d}$, where c_1 is chosen so that the hard instances account for γN of all instances.[3] Figure 3 shows this setting.

3.3 Lower Bound on Hinge Loss

In the simplified case, we assume that the algorithm starts training with the hyperplane $w_0 = w^* = \frac{1}{\sqrt{d}}(1 \ldots 1)$, and keeps it throughout the training, only updating ψ. In reality, each hard instance can be decomposed into a component that is parallel to w^*, and a component that is orthogonal to it. The expected contribution of the orthogonal

[3] See the proof of 0-1 case for a similar construction using the central limit theorem.

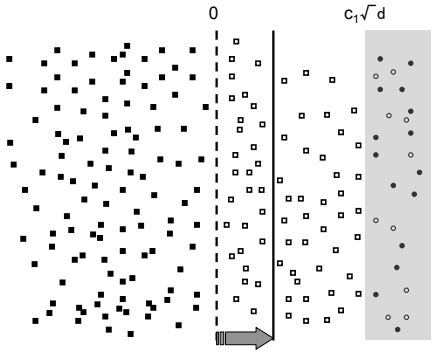

Figure 3: An adversarial case of annotation noise for the voted perceptron algorithm.

component to the algorithm's update will be positive due to the systematic positioning of the hard cases, while the contributions of the parallel components are expected to cancel out due to the symmetry of the hard cases around the main diagonal that is orthogonal to w^*. Thus, while w_t will not necessarily parallel w^*, it will be close to parallel for most $t > 0$. The simplified case is thus a good approximation of the real case, and the bound we obtain is expected to hold for the real case as well.

For any initial value $\psi_0 < 0$ all misclassified instances are labeled -1 and classified as $+1$, hence the update will increase ψ_0, and reach 0 soon enough. We can therefore assume that $\psi_t \geq 0$ for any $t > t_0$ where $t_0 \ll N$.

Lemma 3 *For any $t > t_0$, there exist $\alpha = \alpha(\gamma, T) > 0$ such that $\mathbb{E}(\zeta^2) \geq \alpha \cdot \delta$.*

Proof: For $\psi \geq 0$ there are two main sources of hinge loss: easy $+1$ instances that are classified as -1, and hard -1 instances classified as $+1$. These correspond to the two components of the following sum (the inequality is due to disregarding the loss incurred by a correct classification with too wide a margin):

$$\begin{aligned}\mathbb{E}(\zeta^2) \quad \geq \quad & \sum_{l=0}^{[\psi]} \frac{1}{2^d}\binom{d}{l}\left(\frac{\psi}{\sqrt{d}} - \frac{l}{\sqrt{d}} + \delta\right)^2 \\ & + \frac{1}{2}\sum_{l=c_1\sqrt{d}}^{d} \frac{1}{2^d}\binom{d}{l}\left(\frac{l}{\sqrt{d}} - \frac{\psi}{\sqrt{d}} + \delta\right)^2\end{aligned}$$

Let $0 < T < c_1$ be a parameter. For $\psi > T\sqrt{d}$, misclassified easy instances dominate the loss:

$$\begin{aligned}\mathbb{E}(\zeta^2) \quad \geq \quad & \sum_{l=0}^{[\psi]} \frac{1}{2^d}\binom{d}{l}\left(\frac{\psi}{\sqrt{d}} - \frac{l}{\sqrt{d}} + \delta\right)^2 \\ \geq \quad & \sum_{l=0}^{[T\sqrt{d}]} \frac{1}{2^d}\binom{d}{l}\left(\frac{T\sqrt{d}}{\sqrt{d}} - \frac{l}{\sqrt{d}} + \delta\right)^2 \\ \geq \quad & \sum_{l=0}^{T\sqrt{d}} \frac{1}{2^d}\binom{d}{l}\left(T - \frac{l}{\sqrt{d}} + \delta\right)^2 \\ \geq \quad & \frac{1}{\sqrt{2\pi}}\int_0^T (T + \delta - t)^2 e^{-t^2/2} dt = H_T(\delta)\end{aligned}$$

The last inequality follows from a normal approximation of the binomial distribution (see, for example, Feller (1968)).

For $0 \leq \psi \leq T\sqrt{d}$, misclassified hard cases dominate:

$$\begin{aligned}\mathbb{E}(\zeta^2) \quad \geq \quad & \frac{1}{2}\sum_{l=c_1\sqrt{d}}^{d} \frac{1}{2^d}\binom{d}{l}\left(\frac{l}{\sqrt{d}} - \frac{\psi}{\sqrt{d}} + \delta\right)^2 \\ \geq \quad & \frac{1}{2}\sum_{l=c_1\sqrt{d}}^{d} \frac{1}{2^d}\binom{d}{l}\left(\frac{l}{\sqrt{d}} - \frac{T\sqrt{d}}{\sqrt{d}} + \delta\right)^2 \\ \geq \quad & \frac{1}{2}\cdot\frac{1}{\sqrt{2\pi}}\int_{\Phi^{-1}(\gamma)}^{\infty} (t - T + \delta)^2 e^{-t^2/2} dt \\ = \quad & H_\gamma(\delta)\end{aligned}$$

where $\Phi^{-1}(\gamma)$ is the inverse of the normal distribution density.

Thus $\mathbb{E}(\zeta^2) \geq \min\{H_T(\delta), H_\gamma(\delta)\}$, and there exists $\alpha = \alpha(\gamma, T) > 0$ such that $\min\{H_T(\delta), H_\gamma(\delta)\} \geq \alpha \cdot \delta$ □

Corollary 4 *The bound in theorem 2 does not converge to zero for large N.*

We recall that Freund and Schapire (1999) bound is proportional to $D^2 = \sum_{i=1}^{N} \zeta_i^2$. It follows from lemma 3 that $D^2 = \theta(N)$, hence the bound is ineffective.

3.4 Lower Bound on τ for Voted Perceptron Under Simplified Dynamics

Corollary 4 does not give an estimate on the hard case bias. Indeed, it could be that $w_t = w^*$ for almost every t. There would still be significant hinge in this case, but the hard case bias for the voted forecast would be zero. To assess the hard case bias we need a model of perceptron dynamics that would account for the history of hyperplanes $w_0, \ldots, w_N$ the perceptron goes through on

a sample $x_1, \ldots, x_N$. The key simplification in our model is assuming that w_t parallels w^* for all t, hence the next hyperplane depends only on the offset ψ_t. This is a one dimensional Markov random walk governed by the distribution

$$\mathbb{P}(\psi_{t+1} - \psi_t = r | \psi_t) = \mathbb{P}(x | \frac{y_t - \hat{y}_t}{2} \cdot \langle w^*, x \rangle = r)$$

In general $-d \leq \psi_t \leq d$ but as mentioned before lemma 3, we may assume $\psi_t > 0$.

Lemma 5 *There exists $c > 0$ such that with a high probability $\psi_t > c \cdot \sqrt{d}$ for most $0 \leq t \leq N$.*

Proof: Let $c_0 = F^{-1}(\frac{\gamma}{2} + \frac{1}{2})$; $c_1 = F^{-1}(1-\gamma)$. We designate the intervals $I_0 = [0, c_0 \cdot \sqrt{d}]$; $I_1 = [c_0 \cdot \sqrt{d}, c_1 \cdot \sqrt{d}]$ and $I_2 = [c_1 \cdot \sqrt{d}, d]$ and define $A_i = \{x : v(x) \in I_i\}$ for $i = 0, 1, 2$. Note that the constants c_0 and c_1 are chosen so that $\mathbb{P}(A_0) = \frac{\gamma}{2}$ and $\mathbb{P}(A_2) = \gamma$. It follows from the construction in section 3.2 that A_0 and A_1 are easy instances and A_2 are hard. Given a sample $x_1, \ldots, x_N$, a misclassification of $x_t \in A_0$ by ψ_t could only happen when an easy $+1$ instance is classified as -1. Thus the algorithm would shift ψ_t to the left by no more than $|v_t - \psi_t|$ since $v_t = \langle w^*, x_t \rangle$. This shows that $\psi_t \in I_0$ implies $\psi_{t+1} \in I_0$. In the same manner, it is easy to verify that if $\psi_t \in I_j$ and $x_t \in A_k$ then $\psi_{t+1} \in I_k$, unless $j = 0$ and $k = 1$, in which case $\psi_{t+1} \in I_0$ because $x_t \in A_1$ would be classified correctly by $\psi_t \in I_0$.

We construct a Markov chain with three states $a_0 = 0$, $a_1 = c_0 \cdot \sqrt{d}$ and $a_2 = c_1 \cdot \sqrt{d}$ governed by the following transition distribution:

$$\begin{pmatrix} 1 - \frac{\gamma}{2} & 0 & \frac{\gamma}{2} \\ \frac{\gamma}{2} & 1 - \gamma & \frac{\gamma}{2} \\ \frac{\gamma}{2} & \frac{1}{2} - \frac{3\gamma}{2} & \frac{1}{2} + \gamma \end{pmatrix}$$

Let X_t be the state at time t. The principal eigenvector of the transition matrix $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ gives the stationary probability distribution of X_t. Thus $X_t \in \{a_1, a_2\}$ with probability $\frac{2}{3}$. Since the transition distribution of X_t mirrors that of ψ_t, and since a_j are at the leftmost borders of I_j, respectively, it follows that $X_t \leq \psi_t$ for all t, thus $X_t \in \{a_1, a_2\}$ implies $\psi_t \in I_1 \cup I_2$. It follows that $\psi_t > c_0 \cdot \sqrt{d}$ with probability $\frac{2}{3}$, and the lemma follows from the law of large numbers □

Corollary 6 *With high probability $\tau = \theta(1)$.*

Proof: Lemma 5 shows that for a sample $x_1, \ldots, x_N$ with high probability ψ_t is most of the time to the right of $c \cdot \sqrt{d}$. Consequently for any x in the band $0 \leq v \leq c \cdot \sqrt{d}$ we get $\text{sign}(\langle w^*, x \rangle + \psi_t) = -1$ for most t hence by definition, the voted perceptron would classify such an instance as -1, although it is in fact a $+1$ easy instance. Since there are $\theta(N)$ misclassified easy instances, $\tau = \theta(1)$ □

4 Discussion

In this article we show that training with annotation noise can be detrimental for test-time results on easy, uncontroversial instances; we termed this phenomenon *hard case bias*. Although under the 0-1 loss model annotation noise can be tolerated for larger datasets (theorem 1), minimizing such loss becomes intractable for larger datasets. Freund and Schapire (1999) voted perceptron algorithm and its variants are widely used in computational linguistics practice; our results show that it could suffer a constant rate of hard case bias irrespective of the size of the dataset (section 3.4).

How can hard case bias be reduced? One possibility is removing as many hard cases as one can not only from the test data, as suggested in Beigman Klebanov and Beigman (2009), but from the training data as well. Adding the second annotator is expected to detect about half the hard cases, as they would surface as disagreements between the annotators. Subsequently, a machine learner can be told to ignore those cases during training, reducing the risk of hard case bias. While this is certainly a daunting task, it is possible that for annotation studies that do not require expert annotators and extensive annotator training, the newly available access to a large pool of inexpensive annotators, such as the Amazon Mechanical Turk scheme (Snow et al., 2008),[4] or embedding the task in an online game played by volunteers (Poesio et al., 2008; von Ahn, 2006) could provide some solutions.

Reidsma and op den Akker (2008) suggest a different option. When non-overlapping parts of the dataset are annotated by different annotators, each classifier can be trained to reflect the opinion (albeit biased) of a specific annotator, using different parts of the datasets. Such "subjective machines" can be applied to a new set of data; an item that causes disagreement between classifiers is then extrapolated to be a case of potential disagreement between the humans they replicate, i.e.

[4] http://aws.amazon.com/mturk/

a hard case. Our results suggest that, regardless of the success of such an extrapolation scheme in detecting hard cases, it could erroneously invalidate easy cases: Each classifier would presumably suffer from a certain hard case bias, i.e. classify incorrectly things that are in fact uncontroversial for any human annotator. If each such classifier has a different hard case bias, some inter-classifier disagreements would occur on easy cases. Depending on the distribution of those easy cases in the feature space, this could invalidate valuable cases. If the situation depicted in figure 1 corresponds to the pattern learned by one of the classifiers, it would lead to marking the easy cases closest to the real separation boundary (those between 0 and λ_e) as hard, and hence unsuitable for learning, eliminating the most informative material from the training data.

Reidsma and Carletta (2008) recently showed by simulation that different types of annotator behavior have different impact on the outcomes of machine learning from the annotated data. Our results provide a theoretical analysis that points in the same direction: While random classification noise is tolerable, other types of noise – such as annotation noise handled here – are more problematic. It is therefore important to develop models of annotator behavior and of the resulting imperfections of the annotated datasets, in order to diagnose the potential learning problem and suggest mitigation strategies.

References

Dana Angluin and Philip Laird. 1988. Learning from Noisy Examples. *Machine Learning*, 2(4):343–370.

Beata Beigman Klebanov and Eyal Beigman. 2009. From Annotator Agreement to Noise Models. *Computational Linguistics, accepted for publication*.

Beata Beigman Klebanov, Eyal Beigman, and Daniel Diermeier. 2008. Analyzing Disagreements. In *COLING 2008 Workshop on Human Judgments in Computational Linguistics*, pages 2–7, Manchester, UK.

Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. 1996. A Polynomial-Time Algorithm for Learning Noisy Linear Threshold Functions. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 330–338, Burlington, Vermont, USA.

Xavier Carreras, Llúis Màrquez, and Jorge Castro. 2005. Filtering-Ranking Perceptron Learning for Partial Parsing. *Machine Learning*, 60(1):41–71.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense Tagging of Unknown Nouns in WordNet. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 168–175, Sapporo, Japan.

William Cohen, Vitor Carvalho, and Tom Mitchell. 2004. Learning to Classify Email into "Speech Acts". In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 309–316, Barcelona, Spain.

Edith Cohen. 1997. Learning Noisy Perceptrons by a Perceptron in Polynomial Time. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 514–523, Miami Beach, Florida, USA.

Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 263–370, Philadelphia, USA.

Michael Collins and Brian Roark. 2004. Incremental Parsing with the Perceptron Algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 111–118, Barcelona, Spain.

Michael Collins. 2002a. Discriminative Training Methods for Hidden Markov Hodels: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 1–8, Philadelphia, USA.

Michael Collins. 2002b. Ranking Algorithms for Named Entity Extraction: Boosting and the Voted Perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 489–496, Philadelphia, USA.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Ponnuswami. 2006. New Results for Learning Noisy Parities and Halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–574, Los Alamitos, CA, USA.

William Feller. 1968. *An Introduction to Probability Theory and Its Application*, volume 1. Wiley, New York, 3rd edition.

Yoav Freund and Robert Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296.

Venkatesan Guruswami and Prasad Raghavendra. 2006. Hardness of Learning Halfspaces with Noise. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 543–552, Los Alamitos, CA, USA.

David Haussler. 1992. Decision Theoretic Generalizations of the PAC Model for Neural Net and other Learning Applications. *Information and Computation*, 100(1):78–150.

James Henderson and Ivan Titov. 2005. Data-Defined Kernels for Parse Reranking Derived from Probabilistic Models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 181–188, Ann Arbor, Michigan, USA.

Michael Kearns and Ming Li. 1988. Learning in the Presence of Malicious Errors. In *Proceedings of the 20th Annual ACM symposium on Theory of Computing*, pages 267–280, Chicago, USA.

Michael Kearns, Robert Schapire, and Linda Sellie. 1994. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2):115–141.

Michael Kearns. 1993. Efficient Noise-Tolerant Learning from Statistical Queries. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 392–401, San Diego, CA, USA.

Marvin Minsky and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Mass.

A. B. Novikoff. 1962. On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata*, 12:615–622.

Miles Osborne. 2002. Shallow Parsing Using Noisy and Non-Stationary Training Material. *Journal of Machine Learning Research*, 2:695–719.

Massimo Poesio, Udo Kruschwitz, and Chamberlain Jon. 2008. ANAWIKI: Creating Anaphorically Annotated Resources through Web Cooperation. In *Proceedings of the 6th International Language Resources and Evaluation Conference*, Marrakech, Morocco.

Dennis Reidsma and Jean Carletta. 2008. Reliability measurement without limit. *Computational Linguistics*, 34(3):319–326.

Dennis Reidsma and Rieks op den Akker. 2008. Exploiting Subjective Annotations. In *COLING 2008 Workshop on Human Judgments in Computational Linguistics*, pages 8–16, Manchester, UK.

Frank Rosenblatt. 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, D.C.

Libin Shen and Aravind Joshi. 2005. Incremental LTAG Parsing. In *Proceedings of the Human Language Technology Conference and Empirical Methods in Natural Language Processing Conference*, pages 811–818, Vancouver, British Columbia, Canada.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 254–263, Honolulu, Hawaii.

Paul Viola and Mukund Narasimhan. 2005. Learning to Extract Information from Semi-Structured Text Using a Discriminative Context Free Grammar. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 330–337, Salvador, Brazil.

Luis von Ahn. 2006. Games with a purpose. *Computer*, 39(6):92–94.