Exploiting Labeling Bias in Learning from Positive and Unlabeled Data

Author 1 [1] **Author 2** [1] **Author 3** [1] **Author 4** [1]

Abstract

In many real world problems such as novelty detection, presence/absence inference in ecology, and fake review detection one deals with a small sample of one class with an abundance of unlabeled data. Our goal is to introduce a new method that helps to train a classifier under such strong assumptions about the available dataset for learning. The main idea is to introduce the prior knowledge that the error encountered by the expert who annotates the data is a smooth function of the features of an example, which as we will discuss a natural assumption in many realistic annotation scenarios.

1. Introduction

For a given i.i.d. pair of features and samples $\{(X_i, y_i)\}_{i=1}^N$, one can use many different methods to derive a consistent estimator for the hypothesis class. When the amount of labeled examples is limited, unlabeled data is used to leverage the learning process. Such algorithms are known as Semi-Supervised Learning (SSL) methods. In the context of binary classification, here we are interested in a relaxation, namely that if we have a collection of unlabeled examples/data-points and a collection of labeled examples, but only from one class can we still leverage the learning process and find a reasonable $h \in \mathcal{H}$ that encounters little loss as the number of samples that agents is exposed to increases? This problem in the literature is known as Learning from Positive and Unlabeled data, or PU-Learning for short (Li & Liu, 2005).

To better explain the nature of the problem lets consider an example. Online reviews today play an important role in peoples choice of products. In many cases businesses hire people or create bots to generate fake reviews and ratings. One important problem then is to identify these fake reviews. Indeed such a problem can be phrased as an instance of learning from positive and unlabeled data; in such a case x_i would be the text of the review, y_i would be the true label of data being fake (positive) or not. Despite the fact that there is a scarcity in the amount of available labeled data points, i.e. detected fake reviews, autonomous algorithms are proposed that seem promising to solve this problem (Mukherjee et al., 2013; Jindal et al., 2010; Ott et al., 2013).

Inspired by the seminal work of Elkan and Noto (Elkan & Noto, 2008), in this manuscript we will consider a slightly different (non-traditional) setting for the data-generating process for for an available dataset of positively labeled or otherwise unlabeled data. In this non-traditional setting we assume data is generated according to a joint distribution $p(X, y, l)$. Indeed our sample is generated in triplets x_i, y_i, l_i where $x_i \in \mathcal{X}$ s.t. y_i is observed only if $l_i = 1$ (or equivalently x_i is annotated by the expert) and it is not observed otherwise ($l_i = 0$). Moreover we assume that $l_i = 1$ *only* when $y_i = 1$, which represents the fact that we only observe positively-labeled data points. We will return to these assumptions in a more precise manner later in section 2.

Our goal in this manuscript is to set forth a modeling for PU-learning where we exploit the fact that the labeling process –at least in many cases– is carried out by humans, and this limits the set of functions/models that can represent the labeling process itself. This latter assumption –as we will see– will enable us to do a better classification when we are to learn from only positive and unlabeled data.

2. Brief Literature Review

PU-learning has been discussed under two different names. Under one branch, the problem is known as the "novelty detection" where the task is to identify "novel" examples while very few novel examples and an abundance of the so called "nominal" examples are available. The general approach in these frameworks –even when no novel data is available– is to predict a level set that contains the support of the nominal distribution and to consider examples lying out of this support as novel examples (Schölkopf et al., 2001; El-Yaniv & Nisenson, 2007; Vert & Vert, 2006; Steinwart et al., 2005; Hero, 2007). It is worthwhile to mention that such an approach has been extended to incorporate unlabeled data to leverage learning (Blanchard et al., 2010; Scott & Blanchard, 2009; Liu et al., 2003).

Another set of research relies on "selected completely at random assumption" (or SCAR for short) which we will introduce formally in the next section, but the idea is that the expert *chooses* the positive examples that are going to be annotated, completely at random and independent of their features. This assumption –to the best knowledge of the

authors– was first motivated in the context of text classification when only example texts from one class are available (Denis, 1998; Denis et al., 2002). SCAR assumption has been exploited in many PU-learning algorithms among which (Elkan & Noto, 2008; Du Plessis & Sugiyama, 2014) and (Ward et al., 2009) are a few.

In both of these cases additional assumptions are made, because one can show that in the general setting it is not possible to identify a unique model that generates the data (see e.g. (Elkan & Noto, 2008; Blanchard et al., 2010)).

2.1. PU Learning and SCAR Assumption

Our focus in this work is going to be to introduce a treatment of Elkan et. al's (Elkan & Noto, 2008) method and for that reason here we study this method more thoroughly. The solution of Elkan et. al. (Elkan & Noto, 2008) to the PU-learning is based upon the following assumptions:

(i) **No False Positive (NFP) assumption:** $p(y = 1|l = 1) = 1$. This condition simply means the only labeled examples are positive examples. This is equivalent to assuming there is no false positive labeling by an expert when it comes to finding the positive labels.

(ii) **Selected Completely At Random (SCAR) assumption:** By this assumption mathematically we mean $p(l = 1|y = 1, X = x) = p(l = 1|y = 1)$. To put it otherwise, X is conditionally independent of l, given y. Elkan et. al. (Elkan & Noto, 2008) named this assumption "sampled completely at random assumption", meaning that the set of positive samples that are revealed to the learning algorithm are chosen randomly and identically from the set of all positive examples.

Despite promising mathematical –and sometimes practical– results under assumptions (i) and (ii), it is important to note that assumption (ii) can be highly violated in real-world problems. For example consider that we are interested to find fake reviews on a reviewing platform. Identifying such deceptive reviews can play a very important role in the success of the platform. For example one expects to find more deceptive reviews with full (or high) rating than average or low-rated reviews. So reviews are indeed different in their strength of fakeness when judged by human subjects. Also humans are very biased in types of reviews that they find to be fake (see (Ott et al., 2013) or (Vrij, 2008) for a more explicit study). Another example of the importance of features in labeling process is liked posts on social network platforms. For example a post on Instagram can be liked by a user because there is a cute cat in the picture, or just because the post is made by a celebrity that the user likes, regardless of the image shared by the celebrity. So indeed the liked posts are highly dependant on the content of the post and therefore on the input feature (in this case the image). Therefore, we believe that $p(l = 1|y = 1, X = x)$ is not constant (w.r.t. to X) –which is assumed in SCAR assumption– but rather dependent on X, because as previously mentioned labeling of the samples in many cases is done by human experts and these experts will be sensitive –or said differently, biased towards– specific features of samples.

Both of the assumptions suggested by Elkan et. al. (Elkan & Noto, 2008) (NFP and SCAR) are about the data-generating process. Motivated by the same approach, namely focusing on the data-generating process, the goal of this work is to find assumptions related to this process, under which the learning of $p(y|X)$ would be feasible, even if Assumption (ii) is violated.

In what follows we will also assume Assumption (i) holds. Then, as is shown in (Elkan & Noto, 2008) we have the following:

$$p(l = 1|X = x, y = 1)p(y = 1|x) = p(l = 1|x) \quad (1)$$

Note that all the three terms in (1) are functions of x. For simplicity and as a convention we will refer to these posteriors with the following shorthands interchangeably. We chose $s(x)$ for $p(l = 1|X = x, y = 1)$ so that s stands for the initial letter in "selection process", which as we will discuss represents the selection procedure of "which examples of the positive class to be labeled by the expert?". Similarly we chose $t(x)$ for $p(y = 1|X = x)$ where t is the initial letter of "target" function. Finally we represent $p(l = 1|X = x)$ with $h(x)$. Assumption (ii) therefore is equivalent to $s(x) = c$, where c is a constant independent of the value of X.

3. PU-learning Based On Functional Structures

As we described in section 2, labeling examples in many cases are done by experts, and these experts will be sensitive to what the inputs are, and how well such experts can read a signal from the provided set of features for any given example. So an important question is, under what conditions we are still able to identify s and t in (1) and therefore "de-bias" the labeling process that has been affected by expert bias in choice of features for classification

We claim that by imposing specific structural properties on s, t and h we can still identify s and more importantly t. One way of encoding a difference between s and t is to assume $s(x)$ has a smoother structure compared to $t(x)$. Indeed this assumption makes sense considering that the labeling process by an expert would be more smooth –and

therefore sensitive to the signal to noise ratio in a given example– than an oracle which –almost perfectly– can predict the label for a given example. Let's see an extreme case where such an assumption holds.

3.1. An Extreme Scenario

Let's assume that real data generating process is deterministic, i.e.

Assumption 1. *Assume X is a random variable belonging to the measure space $(\mathcal{X}, \Sigma, \mathbb{P})$ and there exist $\mathcal{A} \in \Sigma$ s.t. $\mathcal{A} \subseteq \mathcal{X}$*

$$\forall x \in \mathcal{A},\ \Pr(y|X = x) = 1,$$

and zero otherwise.

We show that under this condition and with an extra assumption the classification task can be done *almost* perfectly. This second assumption roughly states that there is a similarity between s and t, i.e. the fact that an example is labeled as positive, makes it highly likely to be correctly labeled by an expert.

Assumption 2. *The support of s is very similar to the support of t, i.e.*

$$\Pr\big(\operatorname{supp}(t) \setminus \operatorname{supp}(s)\big) < \epsilon,$$

for some given $\epsilon > 0$.

Define a classifier C_0 as follows:

$$C_0(x) = \begin{cases} 1 & \text{if } \ \mathrm{h(x)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Then we can show the following:

Lemma 1. *Suppose assumptions 1 and 2 hold. Then C_0 has an increase of ϵ in risk compared to Bayes classifier.*

Proof. The proof can be found in supplementary material. □

Despite this interesting theoretical guarantee this case is an extreme scenario with unrealistic assumptions. To remedy this we will introduce a more realistic modeling assumption in the next section.

4. Incorporating Smoothness Assumptions

As we mentioned in the previous section, Assumption 1 is a relatively unrealistic assumptions and something that is not typically assumed in classification problems about the correct hypothesis. Here we replace this assumption and also Assumption 2 in previous section with the following:

Assumption 3. *The labeling process is related to the features smoothly. More precisely we will assume that $p(s = 1|X, y = 1)$ comes from a parametric family of functions, i.e. we are seeking to find the parameter θ_0 where $p(s = 1|X, y = 1, \theta)$ is $s(x)$ and θ captures in some way the smoothness of these families of functions.*

To justify this assumption notice that $p(l = 1|X, y = 1)$ somehow encodes the labeling process for positively labeled data. As we discussed, what Elkan et al. assume (Elkan & Noto, 2008) is that the labeling process itself for positive labels is independent of X and therefore, they implicitly assume there is no "signal" in X that plays a role in the labeling process of positive examples, which means the smooth function above simply is a constant function. Our smoothness assumption means that for an example X with positive label, the labeling expert will be sensitive to the features of an example through a smooth function. Hence if the features of X are slightly changed the probability associated with X being labeled will also slightly change.

To realize this assumption in what follows we will assume $s(x)$ is a sigmoidal function.

4.1. Sigmoidal Product Model (SPM)

A special case to consider is when $s(x)$ and $t(x)$ are both sigmoidal functions, where by sigmoidal function we mean

$$\sigma(x; a, b) = \frac{1}{e^{-ax+b}}.$$

We will call such a family of models as Sigmoidal Product Models (SPMs). We will derive an identifibaility result for this family of models, introduce an algorithm for parameter estimation and eventually we provide empirical examples showing that our model does better in classification compared to PU-learning method of Elkan et. al's (Elkan & Noto, 2008) or compared to the case where h is used as a classifier instead of t, which we call the naïve classifier. To this end assume $X_i \overset{i.i.d.}{\sim} \mathbb{P}$ is given s.t. $X_i \in \mathbb{R}^n$ where $\mathbb{P}$ is an arbitrary distribution. Also assume $\theta_t := (\alpha_t, \beta_t) \in \mathbb{R}^n \times \mathbb{R}$ and $\theta_s := (\alpha_s, \beta_s) \in \mathbb{R}^n \times \mathbb{R}$ represent the parameters of t and s respectively, i.e.

$$p_{\theta_s}(l = 1|X = x, y = 1) = \frac{1}{\exp(-\alpha_s^T x + \beta_s) + 1}, \text{ and}$$
$$p_{\theta_t}(y = 1|X = x) = \frac{1}{\exp(-\alpha_t^T x + \beta_t) + 1}$$

Notice that –as is usually done– we can reparametrize the expressions to include the bias as part of the feature vectors by changing $X_i = (x_{i1}, ..., x_{in})$ vectors to $X_i = (x_{i1}, ..., x_{in}, 1)$. And now these conditionals take the fol-

lowing simpler form

$$p_{\theta_s}(s=1|X=x,y=1)=\frac{1}{\exp(-\theta_s^T x)+1},\ \text{and}$$
$$p_{\theta_t}(y=1|X=x)=\frac{1}{\exp(-\theta_t^T x)+1}$$

In what follows we first show that one advantage of SPMs are their identifiability. This is indeed very interesting because we are dealing with a dataset where we only have positively labeled examples. There is a caveat; the identification of parameters is up to permutation, i.e. we can infer the parameters of SPM but up to a permutation between the parameters belonging to one factor and the ones belonging to the other factor of an SPM ($s(x)$ and $t(x)$).

Lemma 2. *The model based on multiplication of logistics with the conditional likelihood defined in (3) is identifiable up to permutation of θ_t and θ_s, i.e.*

$$\mathcal{L}(\mathbf{X};\theta_t,\theta_s)=\mathcal{L}(\mathbf{X};\theta'_t,\theta'_s)$$

if and only if $\theta'_t=\theta_t$ and $\theta'_s=\theta_s$ or, $\theta'_t=\theta_s$ and $\theta'_s=\theta_t$.

Proof. The proof is omitted and is available in supplementary material. □

Based on this lemma in the next section we introduce an algorithm to learn the parameters of an SPM and also report some results on synthetic and real-world datasets where the ground truth is known about the real labels of the datasets.

5. Experiments

In this section we report the success of SPM used to learn a classifier from positive and unlabeled data only for real-world and synthetic datasets. For this purpose we rely on Algorithm 1, where we minimize the conditional log-likelihood

$$\prod \Pr(l_i|X_i,\theta_s,\theta_t) \tag{2}$$

which is derived in the appendix (See (4)).

5.1. Synthetic Data for SPMs

Based on Lemma 2, we would like to report some empirical results that indeed one can recover the true classifier when the data is generated based on an SPM model, given an extra condition to resolve the ambiguity of permutation of parameters of SPM as described in Lemma 2. For the sake of this experiments two random vectors $\theta_1=(\alpha_1,\beta_1)$ and $\theta_2=(\alpha_2,\beta_2)$ are chosen from a 10-dimensional Gaussian distribution with mean $\mu=0$ and diagonal covariance matrix $25I_{10}$, where I_{10} is the identity matrix of order 10. Then we choose θ_t to be $\theta_i (i=1,2)$ with a larger l_2 norm

Algorithm 1 SPM learning algorithm

`// Estimating` θ_s `and` θ_t

1 Use any optimization method to minimize the negative log-likelihood $\mathcal{L}$ in (2) and recover θ_1 and θ_2, the parameters of the SPM.

`// Choosing the right permutation`

2 Choose whether $\theta_1=\theta_s$ or $\theta_2=\theta_s$ based on condition C. For example motivated by Assumption 1 the condition can be "if $\|\theta_1\|_2<\|\theta_2\|_2$ then $\theta_1=\theta_t$". This is because the sigmoidal function with smaller absolute value of parameters is a steeper one and more "deterministic-like".

`// Returning the classifier`

3 The sigmoidal model with θ_t as its parameter is used for classification.

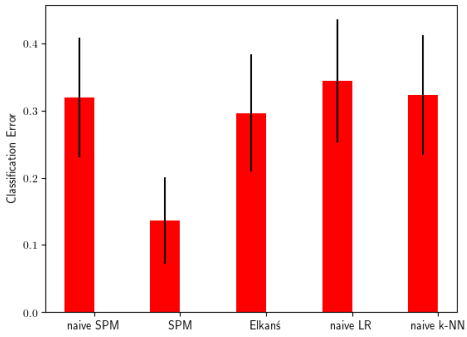

Figure 1. Performance of different methods for learning from positive and unlabeled data. Our proposed method SPM outperforms all the other methods. In the above plot "naive-xx" methods are for cases where classifier xx has been used on training data (X_i,l_i) and tested on (X_i,y_i).

for α_i i.e. $\|\alpha_i\|_2$, and set θ_s to the other chosen random vector. This is because we believe the true classifier as sharper separation of the two classes compared to the selection function. Figure 1 shows the average classification error of Logistic Regression (LR), our proposed method SPM, naïve SPM where we fit the product of sigmoid functions and use this model for classification (the full product), Elkan's method (Elkan & Noto, 2008) when Logistic Regression over 1000 different trials. In general in what follows "naive xx" classifier is a classifier where we use "xx" model to train it on (X_i,l_i) pairs but measure its accuracy on (X_i,y_i) pairs. As it can be seen from the plot SPM and naïve method both outperform Elkan's method. The confidence intervals are calculated using Hoeffding bound on the classification error for each trial.

Before getting to experiments with real data we describe a peculiar method of reporting the test error based on *nested*

cross-validation which has a higher reliability than the traditional train-test split datasets used in machine learning.

5.2. Nested Cross-Validation

Although heavily addressed in other sciences (Varma & Simon, 2006), only is recently reproduciblity becoming an important problem in the field of machine learning up to the point that ICML 2017 has a dedicated workshop on reproducibility[1]. With this in mind reporting the success of a learning model especially when sample sizes are relatively small needs to be done with more care than it is traditionally done by splitting the dataset into training and test set. Indeed this arbitrary splitting can be a big source of bias if the asymmetry by splitting the data in such an arbitrary way is not compensated by an extra effort. Nested Cross-Validation is a method that has two-fold benefits (Varma & Simon, 2006). On one hand it helps us to infer the hyper-parameters of the model thanks to the essence of the algorithm which is cross-validation. But additionally through nested folds –inside the original folds of a cross-validation– it helps one to get an unbiased estimate of the so-called "out-of-sample error" or test error. For these reasons and especially since our dataset size and computational costs are not too high we will be using nested cross-validation to report our test errors in the experiments below.

5.3. Dataset I: Animal/No-Animal Detection

In recent years there is an abundance of datasets on human perception, recognition and assessment on different visual/auditory tasks. Such datasets in many cases can be divided into correct, incorrect, and undecided assessment by humans. Since the ground truth in these cases are known these datasets seem to be a good candidate for our experiments. (Walther et al., 2011) is an example of such a dataset, where humans are asked to categorize drawings that are incomplete. Another such dataset is (Serre et al., 2007) where human object recognition is assessed under image rotations for recognizing animal vs. non-animal patterns. Similar datasets are introduced in (Borji & Itti, 2014).

We focus here on a dataset which is presented in (Delorme et al., 2004)[2]. We briefly describe the experimental paradigm of this dataset. 14 subjects (7 male, 7 female) participated in a study where they are performing a go-nogo categorization task, which is basically equivalent to a classification task. Each subject in the main experiment responded to 2500 stimuli, but here we focus on a sub-task done by the subjects where they had to decide if there is an animal in the shown picture/stimulus or not. There were 10 trials of 100 picture each were the pictures are 50/50 balanced between animal picture and non-animal pictures. During all this time the brain activity of the subjects was recorded using EEG recording technique.

To turn the data collected in this experiment into a proper dataset for our LePU setting we proceeded as follows. For a given example (X, y_{sub}, y_{real}), y_{sub} is the label given by the subject to X and y_{real} is the real label associated with X. X here is the image (animal or not an animal that was shown to the subjects during experiments). Assuming that these labels are binary, we define $l = y_{real}y_{sub}$. This in a way enforces the condition that subjects only classify positive examples. Unfortunately a two-alternative forced choice experimental design is pretty common in psychophysics and neuroscience and finding dataset that subjects are allowed to be indecisive in a binary task is uncommon. Notice that here the data relevant to EEG recordings are discarded.

Because for image classification the dataset is quite small, we used a pretrained neural network known as VGG16 (Simonyan & Zisserman, 2014) as an initial feature extractor for our task. We pass our initial images through this pretrained neural net and take the activation of the first fully connected layer of this network as the feature set for all of our examples. Then, mainly for computational efficiency, we apply a PCA on our new featurized examples to reduce the number of features from 4096 to 50.

After this preprocessing we have applied the SPM method to our featurized dataset using the sample set (X_i, l_i) which we described its construction previously. Then this training set is used infer the posterior $p(y_i = 1|X_i = x_i)$.

Early experiment results confirm that PeLU method proposed by (Elkan & Noto, 2008) indeed fails in this case to be effective even compared to the naïve classification where we use $p(l = 1|X = x)$ to estimate $p(y = 1|X = x)$. To carry out the experiments for this dataset we chose 6 of the subjects (which here we will be referring to with 3-letter names as it is done in the original dataset) for which the classification error for subjects were the highest. This is done, mainly because when human accuracy is really high (say above $\%90$) this implies $s(x)$ is also really high which bounds in it with $0.9 < s(x) < 1.0$. Therefore technically the assumption of Elkan et al. i.e. $s(x)$ being constant holds and their method provides a good estimation. Table **??** shows the estimation of human classification error on

[1] https://icml.cc/Conferences/2017/Schedule?showEvent=16

[2] The dataset itself can be downloaded from https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html

the dataset ($\hat{c}$), and also estimated value of c under Assumption i, i.e. using $c = p(l = 1|, y = 1, X) = p(l = 1|y = 1)$ based on the frequency of occurences in the dataset.

	$\hat{c}$	$\hat{e}$
hth	0.888	0.072
fsa	0.814	0.097
cba	0.954	0.076
ega	0.948	0.064
mta	0.854	0.094
clm	0.922	0.073

Table 1. This table shows the empirical values of $\hat{c}$ and $\hat{e}$ for different subjects on available dataset.

We compare the success of SPM and Logistic Regression and Elkan's method based on these six datasets (for subjects 'hth', 'fsa', 'cba', 'ega', 'mta', 'clm') that are built using the responses of these subjects after being exposed to the stimuli (pictures of animals and non-animals). For regularization we have done a gridsearch over 100 equally distanced points in log-scale between 10^{-4} and 10^4 as the regularization coefficient with l_2 norm both for Elkan and Logistic Regression methods. For SPM method we have used the same range but this time with only 10 equidistant splits in the log-domain, mainly due to processing time constraints. The nested CV is used with 3 inner and 3 outer loops/folds. Tables 2, 3, 4, 5 and 6 report the results for other subjects. These subjects are chosen based on the fact that the estimate value for their $\hat{c} = \sum_{i=1}^{N} \frac{h(X_i)}{s(X_i)}$ was relatively smaller than 1. Table 1 shows the error by humans.

Fold #	SPM	Elkan	Naïve	Real
1	**0.9743**	0.9734	0.9735	0.9745
2	**0.9856**	0.9831	0.9852	0.9873
3	0.9600	0.9827	**0.9841**	0.9866
Avg.	0.9733	0.9797	**0.9809**	0.9828

Table 2. AUC scores for different methods based on data generated from subject cba's perception of pictures. We had 3 outer and 3 inner folds. Naive method outperforms both our method and Elkans's method on average. In the above Naïve means that we used the classifier trained using (X_i, l_i) to label the test dataset. And Real means that we used the real y_i's to get a trained classifier.

Although Naïve method does equally good job in all the cases compared to our method, what is interesting is that Elkan's method never does better than any of the two. This in a way shows how much SCAR assumption is actually violated in real datasets. Also it is important to emphasize that due to speed constraints we did far less splitting of the parameter space for SPM than we did for both Naïve and Elkan's method (10 vs. 100). Hopefully with better optimization we might be able to outperform the Naïve method as well. Also again, it is important to note that for this dataset the estimated value of c based on frequency, i.e. the estimate of $p(l = 1|y = 1, X)$ is around 90% for all of the subjects and such high value already forces $p(l = 1|y = 1, X)$ to be a near-constant function. This might be another reason why our method does not perform particularly well. Still this does not explain why naïve method does better than both of these methods and this needs a further investigation.

Fold #	SPM	Elkan	Naive	Real
1	0.9820	0.9825	**0.9826**	0.9833
2	**0.9846**	0.9803	0.9824	0.9859
3	0.9769	0.9778	**0.9780**	0.9811
Avg.	**0.9812**	0.9802	0.9810	0.9834

Table 3. AUC scores for different methods based on data generated from subject mta's perception of pictures. We had 3 outer and 3 inner folds. As one can see our method outperforms both Elkan and Naive method on average. Also notice that on folds 1 and 3 Naive method outperforms Elkan.

Fold #	SPM	Elkan	Naive	Real
1	0.9884	0.9884	**0.9889**	0.9905
2	**0.9802**	0.9796	0.9798	0.9795
3	0.9841	0.9834	**0.9843**	0.9860
Avg.	0.9842	0.9838	**0.9854**	0.9853

Table 4. AUC scores for different methods based on data generated from subject ega's perception of pictures. We had 3 outer and 3 inner folds. Naive method outperforms both our method and Elkans's method on average.

5.4. Datset II: Detecting Fake Reviews

Today we choose most of the products and services that we use and/or pay for based on the online reviews that other consumers provide for us. Exampels of this are reviews on Amazon, Yelp, TripAdvisor, Airbnb, etc. However writing deceptive fake reviews is a common phenomenon that all these companies need to deal with in order to keep the authenticity of the assessment of the product by the consumers. Additionally finding fake reviews, whether written to promote a product or to devalue another product, is a hard task for humans as it is emphasized in the deception detection literature (Bond Jr & DePaulo, 2006). For this reason developing algorithms for better detection of fake and deceptive reviews is of utmost importance for these industries and also for the consumers. Moreover when a human expert finds a fake review there is enough evidence that led them to believe that is a fake review. However there is no easy way of "proving" that a review is authentic. Therefore for humans, we do have a high precision in finding fake reviews but a low recall.

Fold #	SPM	Elkan	Naive	Real
1	**0.9691**	0.9632	0.9649	0.9729
2	**0.9836**	0.9750	0.9747	0.9896
3	0.9861	**0.9890**	**0.9890**	0.9835
Avg.	**0.9796**	0.9757	0.9762	0.9835

Table 5. AUC scores for different methods based on data generated from subject fsa's perception of pictures. We had 3 outer and 3 inner folds. Our method outperforms both Naïve and Elkans's method on average.

Fold #	SPM	Elkan	Naive	Real
1	**0.9805**	0.9784	0.9803	0.9850
2	0.9812	0.9823	**0.9839**	0.9872
3	0.9806	0.9804	**0.9808**	0.9770
Avg.	0.9808	0.9803	**0.9817**	0.9831

Table 6. AUC scores for different methods based on data generated from subject hth's perception of pictures. We had 3 outer and 3 inner folds. Naive method outperforms both our method and Elkans's method on average.

Suppose we were to incorporate machine learning to remedy the problem of finding fake reviews using a training set collected by annotation done by experts. Due to the nature and hardness of the problem of detecting fake reviews by humans (Ott et al., 2013) we are dealing with a classification (carried out by humans) where procession is high and therefore positively annotated examples have an accurate annotation but since recall is low, negatively annotated examples are highly noisy and have am inaccurate annotation. As a result if we take only the fake reviews as the given initially annotated dataset (which there is a high accuracy over it) we are indeed dealing with a case of learning from positive and unlabeled data only. To see if indeed our methods helps to improve human annotations of reviews we have used the dataset from (Ott et al., 2013). The dataset of (Ott et al., 2013) is available through `http://myleott.com/op-spam.html`[3]. In this dataset 400 fake reviews are generated using Amazon Mechanical Turk. Also 400 5-star reviews for 20 most popular hotels in Chicago are scraped and assumed to be authentically positive reviews, since the claim is that popular places are less prone to be attacked by spammers (Ott et al., 2013). Then three undergraduate students are asked to label all the 800 reviews and identify the fake reviews. Here we will use these human judgments as our input dataset and try to build a classifier based on this dataset that is more accurate in finding fake reviews.

[3]Although three human judges are employed in this study to annotate authentic and fake reviews, their answers are not provided on Myle Ott's website. We have received this dataset through personal communication.

Fold #	SPM	Elkan	Naive	Real
1	**0.9817**	0.9813	0.9816	0.9851
2	0.9873	**0.9882**	0.9879	0.9897
3	**0.9773**	0.9769	0.9768	0.9790
Avg.	**0.9822**	0.9821	0.9821	0.9846

Table 7. AUC scores for different methods based on data generated from subject clm's perception of pictures. We had 3 outer and 3 inner folds. Our method outperforms both Naive method and Elkans's method on average.

To this end, similar to the neuroscience data, we have used the human judgments over the corpus of reviews and mixed them with the ground truth of reviews –which is available– to create a dataset with positive and unlabeled data only (again, notice that similar to neuroscience case we just need to take a logical "and" (or $\wedge$) between the real labels and human judgment of labels to create a list of positively or otherwise unlabeled dataset). The nested CV is used with 5 inner and 5 outer loops/folds.

We compare the success of SPM and Logistic Regression and Elkan's method based on these three datasets that are built using the responses of the three judges that annotated fake reviews. For regularization we have used 1000 equally distanced points in log-scale between 10^{-4} and 10^{4} both for Elkan and Logistic Regression. For SPM method we have used the same range but this time with only 100 equidistant splits in the log-domain. The results for these three subjects are provided in Tables 8, 9 and 10.

Fold #	SPM	Elkan	Naive	Real
1	**0.4510**	0.4157	0.4157	0.8745
2	0.5458	**0.6375**	0.5625	0.8917
3	**0.5830**	0.5385	0.5628	0.8988
4	**0.6508**	0.6429	0.5040	0.9286
5	**0.8083**	0.75	0.8	0.8792
Avg.	**0.6078**	0.5969	0.5690	0.8945

Table 8. AUC scores for different methods based on data generated from subject 1's judgments of fake reviews. We had 5 outer and 5 inner folds. As one can see our method outperforms both Elkan and Naive method on 4 out 5 of folds and also on average. Also notice that on folds 3 and 5 Naive method outperforms Elkan.

It can be seen that in for all three subjects our method (SPM) outperforms Elkan et. al. methods and the natïve classifier in 4 out of 5 folds of nested cross-validation and on average of all the folds. We take that as a strong signal that paying attention to the dependence of selection function on the input, i.e. $s(x)$ on x can contribute to the success of the learning algorithm to learn $p(y|X)$.

Fold #	SPM	Elkan	Naive	Real
1	**0.4196**	0.3882	0.4157	0.8745
2	**0.5625**	0.4542	**0.5625**	0.8917
3	**0.5870**	0.5547	0.5628	0.8988
4	**0.5952**	0.5714	0.5040	0.9286
5	**0.8083**	0.7667	0.8	0.8792
Avg.	**0.5945**	0.5470	0.5689	0.8945

Table 9. AUC scores for different methods based on data generated from subject 2's judgments of fake reviews. We had 5 outer and 5 inner folds. As one can see our method outperforms both Elkan and Naive method on 5 out 5 of folds and also on average. Also notice that on 4 folds out of 5 folds, and on average, Naive method outperforms Elkan's method.

Fold #	SPM	Elkan	Naive	Real
1	**0.4196**	0.4118	0.4157	0.8745
2	**0.5667**	0.5333	0.5625	0.8917
3	**0.6640**	0.5547	0.5628	0.8988
4	0.6032	**0.6071**	0.5040	0.9286
5	**0.8083**	0.7833	0.8	0.8792
Avg.	**0.6123**	0.5780	0.5690	0.8945

Table 10. AUC scores for different methods based on data generated from subject 3's judgments of fake reviews. We had 5 outer and 5 inner folds. As one can see our method outperforms both Elkan and Naive method on 4 out 5 of folds and also on average. Also notice that on folds 4 out of 5 folds, Naive method outperforms Elkan's method.

6. Conclusions and Outlook

In this work we proposed a novel method for learning from a dataset that only contains positive or otherwise unlabeled examples. Most of the previous works in this area take the SCAR assumption for granted, but it is easy to see that this assumption is violated in many real-world problems as we discussed and also depicted through real-world datasets (where the naïve classifier trained by taking unlabeled data as negative examples outperforms a successful method in literature which relies on SCAR assumption).

We proposed to replace this assumption with structural assumptions about the selection procedure of positive examples, i.e. the posterior probability $p(l = 1|X = x, y = 1)$ that a point is selected to be annotated by an expert given that the point actually is a positive example.

The proposed methods were:

- Extremely discrete structure of $p(y = 1|X = x)$
- Sigmoidal structure for $p(y = 1|X = x)$ and $p(l = 1|X = x, y = 1)$

In this work we mainly focused on the second approach and did detailed comparisons of this method with naïve method and Elkan's proposed method.

However we believe replacing the sigmoidal function for selection function $s(x)$ with the psychometric function will be an essential part of future work, and will boost the performance of our proposed parametric setting. Psychometric function (Wichmann & Hill, 2001) has been studied heavily in psychophysical and is assumed to capture well the structure of human decision-making process in psychophysical tasks. We leave the exploration of the success of this method for future work. However optimization for the objective function when the psychometric function is part of the factorization of $p(s = 1|X = x)$ is tricky and needs some further work for a fast and efficient method.

References

Blanchard, Gilles, Lee, Gyemin, and Scott, Clayton. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(Nov):2973–3009, 2010.

Bond Jr, Charles F and DePaulo, Bella M. Accuracy of deception judgments. *Personality and social psychology Review*, 10(3):214–234, 2006.

Borji, Ali and Itti, Laurent. Human vs. computer in scene and object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 113–120, 2014.

Delorme, Arnaud, Rousselet, Guillaume A, Macé, Marc J-M, and Fabre-Thorpe, Michele. Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. *Cognitive Brain Research*, 19(2):103–113, 2004.

Denis, François. Pac learning from positive statistical queries. In *ALT*, volume 98, pp. 112–126. Springer, 1998.

Denis, Francois, Gilleron, Remi, and Tommasi, Marc. Text classification from positive and unlabeled examples. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'02*, pp. 1927–1934, 2002.

Du Plessis, Marthinus Christoffel and Sugiyama, Masashi. Class prior estimation from positive and unlabeled data. *IEICE TRANSACTIONS on Information and Systems*, 97 (5):1358–1362, 2014.

El-Yaniv, Ran and Nisenson, Mordechai. Optimal single-class classification strategies. In *Advances in Neural Information Processing Systems*, pp. 377–384, 2007.

Elkan, Charles and Noto, Keith. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 213–220. ACM, 2008.

Hero, Alfred O. Geometric entropy minimization (gem) for anomaly detection and localization. In *Advances in Neural Information Processing Systems*, pp. 585–592, 2007.

Jindal, Nitin, Liu, Bing, and Lim, Ee-Peng. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1549–1552. ACM, 2010.

Li, Xiao-Li and Liu, Bing. Learning from positive and unlabeled examples with different data distributions. *Machine Learning: ECML 2005*, pp. 218–229, 2005.

Liu, Bing, Dai, Yang, Li, Xiaoli, Lee, Wee Sun, and Yu, Philip S. Building text classifiers using positive and unlabeled examples. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 179–186. IEEE, 2003.

Mukherjee, Arjun, Venkataraman, Vivek, Liu, Bing, and Glance, Natalie S. What yelp fake review filter might be doing? In *ICWSM*, 2013.

Ott, Myle, Cardie, Claire, and Hancock, Jeffrey T. Negative deceptive opinion spam. In *HLT-NAACL*, pp. 497–501, 2013.

Schölkopf, Bernhard, Platt, John C, Shawe-Taylor, John, Smola, Alex J, and Williamson, Robert C. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Scott, Clayton and Blanchard, Gilles. Novelty detection: Unlabeled data definitely help. In *Artificial Intelligence and Statistics*, pp. 464–471, 2009.

Serre, Thomas, Oliva, Aude, and Poggio, Tomaso. A feedforward architecture accounts for rapid categorization. *Proceedings of the national academy of sciences*, 104 (15):6424–6429, 2007.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Steinwart, Ingo, Hush, Don, and Scovel, Clint. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(Feb):211–232, 2005.

Varma, Sudhir and Simon, Richard. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

Vert, Régis and Vert, Jean-Philippe. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7(May):817–854, 2006.

Vrij, Aldert. *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley & Sons, 2008.

Walther, Dirk B, Chai, Barry, Caddigan, Eamon, Beck, Diane M, and Fei-Fei, Li. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23): 9661–9666, 2011.

Ward, Gill, Hastie, Trevor, Barry, Simon, Elith, Jane, and Leathwick, John R. Presence-only data and the em algorithm. *Biometrics*, 65(2):554–563, 2009.

Wichmann, Felix A and Hill, N Jeremy. The psychometric function: I. fitting, sampling, and goodness of fit. *Perception & psychophysics*, 63(8):1293–1313, 2001.

Supplementery Material

6.1. Derivation of Log-likelihood and its Gradient for SPMs

Based on the conditional distributions defined in Section 4, the likelihood function $\mathcal{L}(\mathbf{X}, \mathbf{s}; \theta_t, \theta_s)$ looks as follows:

$$\mathcal{L}(\mathbf{X}, \mathbf{l}; \theta_t, \theta_s) = \prod_{i=1}^{N} \left[\left(\frac{1}{\exp(-\theta_t^T \mathbf{x}_i) + 1} \right) \times \left(\frac{1}{\exp(-\theta_s^T \mathbf{x}_i) + 1} \right) \right]^{l_i} \times \left[\left(1 - \frac{1}{\exp(-\theta_t^T \mathbf{x}_i) + 1} \times \frac{1}{\exp(-\theta_s^T \mathbf{x}_i) + 1} \right) \right]^{(1-l_i)} \tag{3}$$

Now using the notation $\pi_{ti} = (\exp(-\theta_t^T \mathbf{x}_i) + 1)^{-1}$ and $\pi_{si} = (\exp(-\theta_s^T \mathbf{x}_i) + 1)^{-1}$ we get the following conditional negative log-likelihood:

$$-\log(\mathcal{L}(\mathbf{X}, \mathbf{l}; \theta)) = \sum_{i=1}^{N} -\left(\log(\pi_{ti}) + \log(\pi_{si})\right) l_i - (1 - l_i) \log(1 - \pi_{ti}\pi_{si}). \tag{4}$$

Also in what follows we calculate the derivatives of log-likelihood function of this model that has been used in gradient descent algorithm for finding the optimal SPM. Notice the gradient of π_{ti} and π_{si} is as follows:

$$\nabla \pi_{ti} = \mathbf{x}_i \pi_{ti}(1 - \pi_{ti}), \nabla \pi_{si} = \mathbf{x}_i \pi_{si}(1 - \pi_{si}).$$

Based on this we get:

$$\frac{\partial - \log \mathcal{L}(\mathbf{X}, \mathbf{l}; \theta_t, \theta_s)}{\partial \theta_t} = \sum_{i=1}^{N} -l_i \mathbf{x}_i (1 - \pi_{ti}) + \frac{1 - l_i}{1 - \pi_{ti}\pi_{si}} (\pi_{ti}(1 - \pi_{ti})\pi_{si}) \mathbf{x}_i = -\sum_{i=1}^{N} (1 - \pi_{ti}) \Big[l_i - \frac{\pi_{ti}\pi_{si}(1 - l_i)}{1 - \pi_{ti}\pi_{si}} \Big] \mathbf{x}_i$$

6.2. Proof of Lemma 1

Proof. Recall that for a given joint distribution $\Pr(X, Y)$, the Bayes classifier is defined as follows:

$$C_B(x) = \begin{cases} 1 & \text{if } \ \mathrm{f_B(x)} > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

where $f_B = \mathbb{E}(Y = 1|X) = \Pr(Y = 1|X)$. Now the risk of any arbitrary classifier for any given $X = x$ is:

$$\begin{aligned} &\Pr(Y \neq C(X)|X = x) = 1 - \\ &[P(Y = 1, C(X) = 1|X = x) + \\ &\Pr(Y = 0, C(X) = 0|X = x)] \\ &= 1 - [C(x) f_B(x) + (1 - C(x))(1 - f_B(x)] \\ &= f_B(x) + (1 - 2f_B(x))C(x). \end{aligned}$$

Define $R_B(x)$ and $R_0(x)$ as the conditional risk values associated with these two classifiers, i.e. $R_0(x) = P(Y \neq C_0(x)|X = x)$ Taking the difference between the conditional risk of introduced classifier C_0 and C_B we get:

$$\begin{aligned} \Delta R(x) := &\Pr(Y \neq C_0(X)|X = x) - \\ &\Pr(Y \neq C_B(X)|X = x) = \\ &(2f_B(x) - 1)(C_B(x) - C_0(x)). \end{aligned}$$

Notice that if $x \notin \mathcal{A}$, then $\Delta R(x) = 0$. This is indeed the case because if $x \notin \mathcal{A}$ then $f_B(x) = C_B(x) = t(x) = 0$, which implies $h(x) = 0$ and therefore $C_0(x) = 0$ and so $\Delta R(x) = 0$. And if $x \in \mathcal{A}$ then $f_B(x) = C_B(x) = 1$ and therefore

$$\Delta R(x) = 1 - C_0(x).$$

But note that

$$\begin{aligned} \mathbb{E}[R_0] - \mathbb{E}[R_B] = \mathbb{E}_X(\Delta R) = \Pr(x \in \mathcal{A}) \times 0 + \\ \Pr(x \in \mathcal{A}) \times (1 - C_0(x)) \end{aligned}$$

where expectations are taken with respect to random variable X, and according to (2) $1 - C_0(x) \leq \epsilon$ which implies

$$\mathbb{E}[R_0] - \mathbb{E}[R_B] \leq \Pr(x \in \mathcal{A})\epsilon \leq \epsilon$$

□

Proof of Lemma 2

Proof. The "only if" direction is trivial due to symmetry of t and s. For "if" direction, notice that for fixed θ's these two factors are both functions of just $\mathbf{X}$. So without loss of generality if we show that for one datapoint the above equality implies identifiablity, the proof also follows for the case with more datapoints. Also without loss of generality if we can show the proof for one-dimensional case, we can conclude it for multidimensional case as one can set $\mathbf{x} = (x_1, ..., x_n, 1)$ to $\mathbf{x} = (1, 0, ..., 0, 1)$. Finally WLOG we can assume $s = 1$, because the case $l = 1$ has a similar likelihood to $l = 0$, i.e. $Ł(\mathbf{x}, l = 1; \theta_t, \theta_s) = 1 - Ł(\mathbf{x}, l = 0; \theta_t, \theta_s)$. Now assume for a given θ_t, θ_s, θ'_s and θ'_t we have the following:

$$Ł(X, l; \theta_t, \theta_s) = Ł(X, l; \theta'_t, \theta'_s).$$

So we have

$$\frac{1}{\exp(-a_t x + b_t) + 1} \times \frac{1}{\exp(-a_s x + b_s) + 1} = \frac{1}{\exp(-a'_t x + b_t)) + 1} \times \frac{1}{\exp(-a'_s x + b_s) + 1}.$$

Therefore

$$(\exp(-a_t x + b_t) + 1) \times (\exp(-a_s x + b_s) + 1) = (\exp(-a'_t x + b_t)) + 1) \times (\exp(-a'_s x + b_s) + 1).$$

simplifying we get

$$e^{-(a_t+a_s)x+b_s+b_t} + e^{-a_t x+b_t} + e^{-a_s x+b_s} = \quad (5)$$
$$e^{-(a'_t+a'_s)x+b'_s+b'_t} + e^{-a'_t x+b'_t} + e^{-a'_s x+b'_s} \quad (6)$$

Take

$$f(x) := e^{-(a_t+a_s)x+b_s+b_t} + e^{-a_t x+b_t} + e^{-a_s x+b_s} - e^{-(a'_t+a'_s)x+b'_s+b'_t} - e^{-a'_t x+b'_t} - e^{-a'_s x+b'_s}.$$

According to (6) we have $f(x) = 0$ for any $x \in \mathbb{R}$. Also taking derivative of f w.r.t. x we get

$$f'(x) = -(a_t + a_s)e^{-(a_t+a_s)x+b_s+b_t} - a_t e^{-a_t x+b_t} - a_s e^{-a_s x+b_s} + (a'_t + a'_s)e^{-(a'_t+a'_s)x+b'_s+b'_t} + a'_t e^{-a'_t x+b'_t} + a'_s e^{-a'_s x+b'_s} = 0 \quad (7)$$

for any $x \in \mathbb{R}$. We divide the proof into cases.

(i) $a_t > 0$ and $a_s > 0$: This means $a_t + a_s > 0$. It follows that $a'_t \geq 0$ and $a'_s \geq 0$, as otherwise taking the limit of x to infinity leads to a contradiction as right hand side of (6) goes to infinity whereas left hand side of it approaches to a real value. This means $a'_s + a'_t \geq \max\{a'_s, a'_t\}$. But this implies $a_s + a_t = a'_s + a'_t$ as otherwise there will be a dominating exponent in $f(x)$ and as a result

$$\lim_{x\to-\infty} f(x) = \infty \text{ or } \lim_{x\to\infty} f(x) = -\infty$$

which is a contradiction since $f(x) = 0$. Now note it can't be the case that $a'_s = 0$ or $a'_t = 0$. Suppose to the contrary that this is the case. WLOG assume $a'_s = 0$. Divide both sides of (6) with $\exp(a_t + a_s)x$ and take the limits to $-\infty$. From (6) we get

$$e^{b_s+b_t} = e^{b'_s+b'_t} + e^{b'_t} \quad (8)$$

and from (7) we get

$$-a'_t e^{-a'_t x+b_s+b_t} - a_t e^{-a_t x+b_t} - a_s e^{-a_s x+b_s} + a'_t(e^{-a'_t x+b'_s+b'_t} + e^{a'_t x+b'_t}) = 0$$

Now setting $x = 0$ gives

$$-a'_t e^{b_s+b_t} - a_t e^{b_t} - a_s e^{b_s} + a'_t(e^{b'_s+b'_t} + e^{b'_t}) = 0$$

and due to (8) we get

$$-a_t e^{b_t} - a_s e^{b_s} = 0$$

which leads to a contradiction. So we do have $a'_s + a'_t > \max\{a'_s, a'_t\}$. This implies $b_t + b_s = b'_t + b'_s$; this follows by dividing both sides of (6) by $e^{-(a_t+a_s)x}$ and taking the limit $x \to +\infty$. Therefore we get

$$e^{-a_t x+b_t} + e^{-a_s x+b_s} = e^{-a'_t x+b'_t} + e^{-a'_s x+b'_s}$$

now if $a_s \neq a_t$, the dominating term on both sides should be equal with the similar reasoning we did for (6). As a result $a_s = a'_s$ and therefore $a_t = a'_t$ (or $a_s = a'_t$ and therefore $a_t = a'_s$). In either case similar to the proof for 6 it follows that $b_t = b'_t$ and therefore $b_s = b'_s$ (or $b_s = b'_t$ and therefore $b_t = b'_s$). This completes the proof for this case.

(ii) $a_t > 0$ and $a_s = 0$: Similar to what has previously shown, it follows that $a'_t \geq 0$ and $a'_s \geq 0$. Now note that it can't be the case that $a'_t > 0$ and $a'_s > 0$, as we argued in case (i) that this is impossible. So $a'_t = 0$ or $a'_s = 0$. In either case it follows that $a'_s = a_t$ and $a'_t = a_t$ respectively. It follows immediately that $b_t = b'_t$ and therefore $b_s = b'_s$ (or $b_s = b'_t$ and therefore $b_t = b'_s$).

(iii) $a_t = a_s = 0$: Notice that in this case it is obvious that rhs of (6) need also to be independent of x which implies $a'_t = a'_s = 0$. But note that for any non-zero element of either of θ_t or θ_s one can conclude that $b_t = b'_t$ and therefore $b_s = b'_s$ (or $b_s = b'_t$ and therefore $b_t = b'_s$). Unless all the elements of θ_t and θ_s are zero, in which case the likelihood does not depend on the data which is a contradiction.

(iv) $a_t > 0$ and $a_s < 0$: Multiply both sides of (6) with $e^{a_s|x|}$ and we get

$$e^{-(a_t+(a_s+|a_s|))x+b_s+b_t} + e^{-(a_t+|a_s|)x+b_t} + e^{-(a_s+|a_s|)x+b_s} = \quad (9)$$
$$e^{-(a'_t+(a'_s+|a_s|))x+b'_s+b'_t} + e^{-(a'_t+|a_s|)x+b'_t} + e^{-(a'_s+|a_s|)x+b'_s} \quad (10)$$

but now the claim of the lemma follows by reapplying (i), (ii) and (iii) to (10) with x exponents as $a_s + |a_s|$, $a_t + |a_s|$, $a_s + |a_s|$, $a'_s + |a_s|$, $a'_t + |a_s|$ and $a'_s + |a_s|$.

(v) $a_t < 0$ and $a_s < 0$: This case follows by replacing x with $-x$ on both sides of (6). This completes the proof of the lemma.

□