MANIFOLD REGULARIZATION FOR SEMI-SUPERVISED SEQUENTIAL LEARNING

Yvonne Moh *Joachim M. Buhmann*

Department of Informatics, ETH Zurich

ABSTRACT

The sequential data flux in many time-series applications require that only a small fraction of the data are stored for future processing. Furthermore, labels for these data are possibly sparse and they might show some biases. To support learning under such restrictive constraints, we combine manifold regularization with sequential learning under a semi-supervised learning scenario. The online learning mechanism integrates a regularization based on the data smoothness assumptions. We present a proof-of-concept for illustrative toy problems, and we apply the algorithm to a real-world sparse online classification task for music categories.

***Index Terms*—** Online Learning, Semi-Supervised Learning, Classifier Adaptation

1. INTRODUCTION

We address the problem of semi-supervised sequential learning, which is frequently encountered in real world scenarios. Streams of sequential time data are readily accessible. However, procuring labels for these streams of data is often expensive and, in many applications, labels are obtained for only a subset of data. We consider the scenario where the learning algorithm cannot actively seek the class information as in active learning algorithms [1].

Labels are for instance dependent on user behavior, and, hence, are not necessarily i.i.d. drawn. As such, these sparse labels may contain biases, e.g., an experimental selection bias filters out labeled data that only occupy sub regions of the feature space and, therefore, they are not equally distributed over the entire space.

We briefly outline the general learning setup: Data points are available one at a time, with each observation serving first as a test point, and then as a training point. For an incoming data point, a prediction is made. After prediction, the true label *may or may not* be obtained, and the observation is used to update the model in the next step.

In a specific potential application, we envision the online-adaptation of models in personalized hearing aids. Modern hearing aids have an associated classifier that enables the hearing aid to adjust to the acoustic environment, thereby, providing better hearing comfort and clarity [2]. For instance, the user might appreciate piano music, yet he may dislike string instruments. For such a user, the hearing aid should amplify piano music while suppressing violins and string instruments. The scenario of biased labels (occupying specific subspaces) readily arises when the feedback coincide with segments where the user has the opportunity to manually provide labels. For instance, it is unlikely for a user to provide feedback whilst driving a car.

2. BACKGROUND

We consider two machine learning components: sequential online learning [3], and semi-supervised learning [4, 5]. We highlight relevant algorithms and provide some basic details.

Traditional sequential online-learning algorithm follow the cycle of prediction, reward/loss, and learning. At time step t, the algorithm obtains a data point x_t, for which it is required to make a prediction $f_t(x_t)$ using its current model or prediction function f_t. Upon completion, the true label $y_t \in \{-1, +1\}$ is provided, which is used to evaluate the system. The algorithm then exploits the availability of this label to update the model before classifying the next point x_{t+1} using the new prediction function f_{t+1}. In this work, we consider functions chosen from the Reproducing Kernel Hilbert Space (RKHS). $\mathcal{H}_K$ is the associated RKHS of functions for a Mercer kernel $K : X \times X \to \mathbb{R}$, with the corresponding norm $|| \, ||_K$.

We modify this setting by providing labels for a limited subset of data points during the learning phase. The label set $y_t \in \{-1, 0, +1\}$ is extended such that unlabeled data are denoted by $y_t = 0$. A conservative policy for parameter estimation based on such a data stream is to filter out and exclusively evaluate the labeled data. The unlabeled data ($y_t = 0$) is completely ignored.

However, unlabeled data may provide additional information on the data manifold that is not available in the set of labeled data. The structure of the manifold often contains information which is relevant for the labels, e.g., densely populated and connected data spaces often belong to the same class. The prior assumption of smooth data manifolds also suggests that "nearby" datapoints are more likely to share the same labels, while datapoints that are "far" apart are less likely to own the same label. In our paper, this criterion is estimated empirically at time t by a sliding window of the τ most recent data $X_\tau = \{x_{t-\tau+1}, ..., x_t\}$. We impose this constraint of limited memory that only the τ most recent data points can be stored. This constraint holds for practical systems with limited memory.

2.1. Passive-Aggressive Online algorithms

For online learning, we employ the Passive Aggressive Online Algorithms (PA) [6]. PA has been defined for fully supervised scenarios. It solves the constrained optimization problem:

$$f_{t+1} = \arg\min_{f \in \mathcal{H}_K} \|f - f_t\|_K^2 \tag{1}$$

under the constraints that hinge loss vanishes. The hinge loss is given by:

$$l(f; (x, y)) = \begin{cases} 0 & yf(x) \geq 1 \\ 1 - yf(x) & \text{otherwise.} \end{cases} \tag{2}$$

The solution to the convex optimization problem yields a closed form solution which can be expressed in the form:

$$f_{t+1}(x) = f_t(x) + \frac{l(f_t(x_t); (x_t, y_t))}{K(x_t, x_t)} y_t K(x_t, x) \tag{3}$$

Consider the linear case where the prediction function $f_t(x_t) = w_t' x_t$. Geometrically, the update solution w_{t+1} is the vector nearest to w_t which attains a *hinge loss* of zero on the current example x_t (Fig.1, left).

2.2. Manifold smoothness

The goal is to incorporate additional information about the geometric structure of the data distribution. This information is useful if there is a connection between the data distribution and the conditional distribution of the labels $P(y|x)$. Specifically, if two points x_1 and x_2 are close, then the conditional distributions $P(y|x_1)$ and $P(y|x_2)$ are also similar. We can interpret this as a label smoothness constraint [7].

This constraint can be incorporated into the objective function as a regularization term. The data takes a graph representation, where similarities between datapoints are coded via an affinity matrix W, i.e., W_{ij} denotes the similarity between two data points x_i and x_j. When optimizing the prediction function f, the empirical formulation for the smoothness constraints takes the form

$$\min_{f \in \mathcal{H}_K} \sum_{i,j} (f(x_i) - f(x_j))^2 W_{ij}. \quad (4)$$

In this form, the quadratic difference between the labels is weighted by the similarities of the data points. This can be rewritten to a similar form

$$\min_{f \in \mathcal{H}_K} f' L f \quad (5)$$

which uses e.g. the normalized graph Laplacian $L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. Here, D is the diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

3. MANIFOLD REGULARIZATION FOR ONLINE LEARNING WITH SPARSE LABELS

We consider manifold regularization for online learning with sparse labels by directly incorporating the manifold smoothness constraints into the objective function. We specifically consider this extension for the PA-algorithm.

3.1. Manifold regularized PA learning

We extend the PA objective function, by incorporating a regularization term that is defined over a window $X_\tau = \{x_{t-\tau+1}, ..., x_t\}$ of past data. The concept of using such a window is common in online-learning systems that handle concept drifts (e.g. [8]).

In the general formulation, we have

$$f_{t+1} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{2} ||f - f_t||_K^2 + \frac{\gamma}{2} f' L f, \quad (6)$$

such that $1 - y_t f_t(x_t) = 0$. Here, L is the normalized graph Laplacian computed over X_τ.

We can show that the Representer Theorem is valid here and write

$$f^*(x) = \sum_{x_i \in X} \alpha_i^* K(x, x_i). \quad (7)$$

Furthermore, by construction, we have $f_t(x) = \sum_{x_j} \rho_j K(x, x_j)$ where x_j's represent previously seen data and ρ_j represent scalar weights. The solution can be seen as an update of weights plus the uptake of newly seen datapoints.

For simplicity, we consider the linear formulation. Eq 8 below shows Eq. 6 in the linear form, where now the regularization term Eq. 5 is also reformulated for the linear form:

$$w_{t+1} = \arg\min_{w \in \mathbb{R}^n} \frac{1}{2} ||w - w_t||^2 + \frac{\gamma}{2} w' (X_\tau' L X_\tau) w \quad (8)$$

with a single feasible affine constraint $l(w; (x_t, y_t)) = 0$. The regularization term is $\frac{\gamma}{2} w' (X_\tau' L X_\tau) w$, where $\gamma \geq 0$ is a regularization parameter for the manifold smoothness assumption.

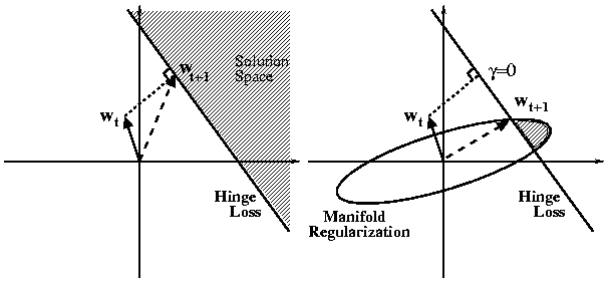

Fig. 1. Illustration of the optimization problem for PA (left). Extension of the optimization for manifold regularization (right).

In PA, the update step at time $t+1$ finds the w nearest to the current hyperplane w_t that induces hinge loss of 0 on the datapoint x_t. The manifold regularization takes the form of a ellipsoid parametrized by γ, and the solution space is now constrained to lie also within the ellipsoid. Hence, w_{t+1} is pushed towards the regularizing ellipsoid, as shown in Fig. 1.

3.2. Convex Optimization

Fortunately, Eq. 8 is convex: The second term depends on the normalized graph Laplacian, which is symmetric positive semi-definite, and hence convex We have the sum of two convex functions evaluated over the same domain, which is also a convex function. Therefore, we are optimizing a convex optimization function under a feasible affine constraint. This fulfills the Slater's conditions, hence optimization is equivalent to satisfying the Karush-Khun-Tucker conditions, and we can solve to obtain (for the linear formulation)

$$w_{t+1} = A(w_t - \lambda y_t x_t), \quad (9)$$

where $\lambda = \frac{1 - y_t x_t' A w_t}{x_t' A x_t}$ and $A = (I + \gamma X_\tau' L X_\tau)^{-1}$. The solution is a transformation of the original solution represented by some small weighted factor of the spectral cluster solution away from the identity I. For $\gamma = 0$, the solution reverts to the original PA.

3.3. Algorithm

The Laplacian Passive Aggressive algorithm (`LapPA`) is presented in Algorithm 1. `LapPA` differs from `PA` in the updates: In `LapPA`, manifold regularization is performed and controlled by the regularization constant $\gamma \geq 0$. The estimation of the Laplacian L necessary for computing the regularization term is performed on the most recent window of data $X_\tau = \{x_{t-\tau+1}, ..., x_t\}$ of fixed size τ. The algorithm is formulated for the linear case. The kernelized version is obtained by replacing all inner products by a general Mercer kernel.

Algorithm 1 `LapPA`

Require: Input: $w_1, \gamma \geq 0, \tau$

initial classifier w_1, regularization constant γ, window size τ

for t = 1, 2, ... **do**
  Predict: $\hat{y}_t = \text{sign}(w_t' x_t)$ for current data x_t
  if y_t available: **then**
    Compute loss $l_t = \max\{0, 1 - y_t(w_t' x_t)\}$
    if $l_t > 0$ **then**
      Update

$$\lambda_t = \frac{1 - y_t x_t'(I + \gamma X_\tau' L X_\tau)^{-1} w_t}{x_t'(I + \gamma X_\tau' L X_\tau)^{-1} x_t}$$

$$w_{t+1} = (I + \gamma X_\tau' L X_\tau)^{-1}(w_t + \lambda_t y_t x_t)$$

    end if
  end if
end for

4. EXPERIMENTAL RESULTS

We demonstrate the behavior of the algorithm `LapPA` on synthetic problems, that serve as a proof-of-concept, and on music data. The linear-separable 2-rods problem examines the algorithm in the linear case. We present a similar illustration for the kernelized version, where two non-linearly separable half moons have to be classified as a test case for semi-supervised learning algorithms. Finally we show the results on music data.

In our experiments, we compare `LapPA` (regularization) to the baseline `PA` where learning is executed only on the labeled data. For the data simulation, 8000 datapoints are drawn i.i.d. and presented in a sequential order. Each experimental setup is based on 100 independent simulations. $p\%$ of the datapoints that are sampled from the selected segments (biases) have accompanying label information, are supplied to the algorithm after the prediction cycle of that datapoint. All initial classifiers f_1 are set at 0.

The graph affinity matrix is calculated over the specified window of the τ most recently seen data. We use the similarity measure $d(x_i, x_j) = \exp(-||x_i - x_j||^2)$, and threshold to keep only the 30% top entries (ϵ-neighborhood graph [9]), and set the remaining edges to 0.

We evaluate using the cumulative accuracy. For a data point x_t, the corresponding predicted dichotomy is given by $\hat{y}_t = \text{sign}(f_t(x_t))$, where f_t is the classifier obtained after processing x_{t-1}. Hence, $Acc = \frac{1}{T}\sum_{t=1}^{T} \delta(\hat{y}_t, y_t)$, where δ is the Kronecker delta.

4.1. Linear classifier

We consider a simple linearly separable problem. The data is constructed from two rectangular regions of classes -1 and $+1$. Each rectangle is partitioned along the horizontal axis into 5 equal segments. Sparse labels ($p\%$ of data) are provided for only one segment for each rod.

Figure 2 shows the final separating hyperplane after sequential learning with labels provided only for a few points sampled from opposite corners of the rectangular regions. Without regularization, `PA` finds a maximum margin solution between the sub-clusters of labeled points. However it ignores the information of the data distribution and erroneously intersects with both regions. This is contrasted by the behavior of `LapPA` which forces the hyperplane away from the solution that cuts through an area of high data density of

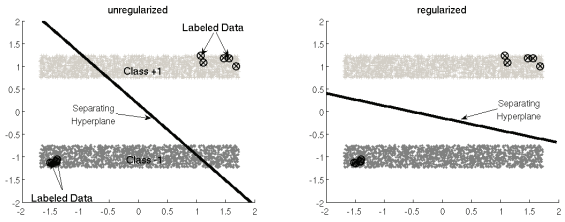

Fig. 2. The black bull-eye markers indicate the sparse, biased labeled datapoints. The separating hyperplane obtained after learning is shown for `PA` (left) and `LapPA` (right).

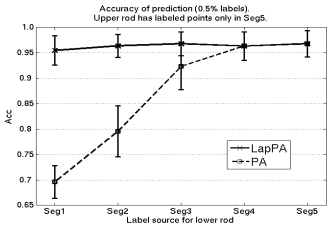

Fig. 3. Recognition accuracy when labels come from respective segments of the rods.

unlabeled data points. Due to the smoothness assumption, it adjusts the algorithm to take a solution that separates the rods even at data regions where no labels were observed.

Figure 3 shows the results when the labels are sampled from different areas of the lower rectangle while only sampling from the right end of the upper bar. When the labeled segments are far apart, `PA` tilts the separating hyperplane to accommodate for a maximum margin separation based entirely on the labeled data, resulting in decreased accuracy. When the two segments are at two differing extreme ends (i.e. Fig 2), the accuracy drops to yield 0.7 with `PA`, whilst `LapPA` still manages to retain the accuracy at over 0.95.

4.2. Kernel classifier

We analyse the two-moons problem (see Fig. 4) often considered as a proof-of-concept problem in the semi-supervised learning community. We define five, evenly distributed segments on each moon, and evaluate the classification performance on biased label sources. For

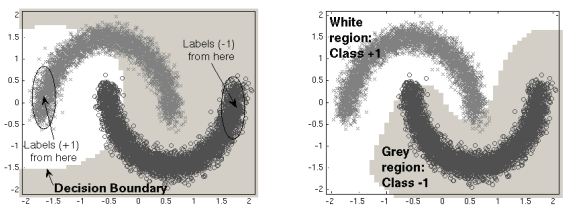

Fig. 4. Class boundaries for 2-moons: with class +1 unshaded, class -1 shaded grey. `LapPA` (right) shows better class separation than `PA` (left).

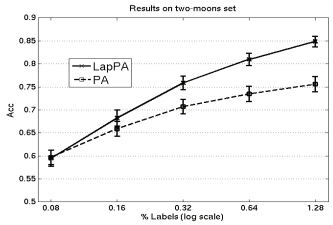

Fig. 5. Accuracy rates for PA and LapPA at different p%.

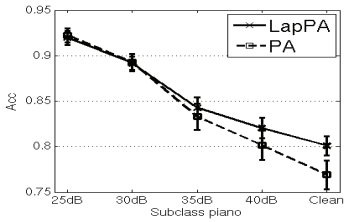

Fig. 6. Results on music data with varying amounts of label information provided. String Quartets at SNR 25dB

learning, we employ RBF kernels.

Figure 4 shows the effect of PA and LapPA. Unregularized learning PA fails to capture the extended source of the labels due to inhomogeneous sampling. In the regularized case, the decision boundary is improved, and the decision boundary is able to track a huge proportion of the information provided by the data density.

In Fig. 5 we see the effect of amount of labels provided. LapPA consistently outperforms PA. Since the PA-algorithm only updates when a loss on a labeled data is incurred, too few labels (e.g. p=0.08% labels $\approx$ 6 labeled points) lead to too few update steps. As such both algorithm show equally poor performance and accuracy is low. When sufficient learning opportunities arise, both algorithms have more opportunities to update the model, and LapPA has the opportunity to incorporate the manifold information at each update, resulting in better performance compared to PA.

4.3. Music dataset

We evaluated the algorithm on a music data set. The music data set is composed of music audio files of two similar classes: piano and string quartets. We artificially added Gaussian white noises to the audio files at different signal to noise ratios (SNR) of 25, 30, 35 and 40 dB and the noise free case. We subcategorize the data according to the noise that were added to the data.

Features were extracted from the 16hHz audio files. We computed spectral centroid, spectral roll, spectral flux, time domain zero crossings and low-energy. Reference [10] provides a good description of all features used. Furthermore, 13 Mel-Frequency Cepstral Coefficients (MFCC) are extracted with a window size of 32ms, with overlapping windows of 16ms. For all features[1], we computed means and variances over 0.8s segments. In a second data reduction step, the means of these statistics over 25 segments yield a 35 dimensional feature vector for each non-overlapping 20 second segment. Linear Discriminant Analysis was then performed on the 10 sub-classes. The entire dataset of 1600 songs is represented by a 19080 datapoints, i.e., an average 12 feature vectors (datapoints) per song.

The songs were randomly shuffled and presented in a stream to the algorithm. Feature vectors were presented in the order of their occurrence in the song. Furthermore, no song-boundary information is provided to the algorithm, i.e., information concerning which feature vectors belong together is not provided. As a consequence, the data stream is not composed of i.i.d. samples.

Figure 6 shows the results of our simulations for the linear form of LapPA with $p = 10\%$. Using LapPA brings a significant improvement over PA when labels are sparse and biased.

[1]Except low-energy, where only means were used

5. CONCLUSIONS

We derived an update solution for sequential online learning with manifold regularization applied to the Passive-Aggressive algorithm. On toy data, regularization improves the cumulative accuracy, when data fulfil the manifold smoothness constraints.

In our experiments, we did not integrate concept drifts, and it will be an extension to investigate the effects of concept drift to these problems. This will also indicate a potential need for adaptive corrections for the hyperparameters such as the window size τ and the strength of regularization γ.

6. REFERENCES

[1] Simon Tong and Daphne Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[2] M. Büchler, S. Allegro, S. Launer, and N. Dillier, "Sound classification in hearing aids inspired by auditory scene anlysis," *Journal of Applied Signal Processing*, vol. 18, pp. 2991–3002, 2005.

[3] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning and games*, Cambridge University Press, 2006.

[4] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, MIT Press, 2006.

[5] Xiaojin Zhu, "Semi-supervised learning literature survey, technical report 1530," Tech. Rep., Department of Computer Sciences, University of Wisconsin, Madison, 2005.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Schwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[7] M. Belkin, P. Niyogi, and V. Sinhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2006.

[8] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Journal Machine Learning*, vol. 23, no. 1, pp. 69–101, April 1996.

[9] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec 2007.

[10] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, 2002.