CONSTRAINED BAYESIAN ACTIVE LEARNING OF A LINEAR CLASSIFIER

Anestis Tsakmalis, Symeon Chatzinotas and Björn Ottersten

SnT - securityandtrust.lu, University of Luxembourg

ABSTRACT

In this paper, an on-line interactive method is proposed for learning a linear classifier. This problem is studied within the Active Learning (AL) framework where the learning algorithm sequentially chooses unlabelled training samples and requests their class labels from an oracle in order to learn the classifier with the least queries to the oracle possible. Additionally, a constraint is introduced into this interactive learning process which limits the percentage of the samples from one "unwanted" class under a certain threshold. An optimal AL solution is derived and implemented with a sophisticated, accurate and fast Bayesian Learning method, the Expectation Propagation (EP) and its performance is demonstrated through numerical simulations.

Index Terms— Active Learning, Constrained Dynamic Programming, Expectation Propagation

1. INTRODUCTION

In many learning scenarios, data are not delivered passively to the learner but they have to be obtained interactively. In this proactive approach, the learner sequentially designs questions and queries an oracle in order to gain consecutively more and more informative answers about a learning problem. In the Machine Learning regime, this special class of learning problems is called *Active Learning* (AL). One of the first AL pieces of work which had a great impact in the research community and dealt with actively identifying regressors is [1]. Later on, the Cognitive Radio research community managed to connect the AL problem with Cutting Plane Methods (CPMs) which basically tackled active classification problems [2, 3]. The same approach was followed by the authors of [4] who also applied CPMs to compression schemes.

A complementary approach to the AL problem is based on its sequential nature. This iterative rationale is connected to the Bayesian Experimental Design framework [5] which in its turn is related to the theory of optimal Decision Making. Researchers from the Decision Making field have exploited a Dynamic Programming (DP) approach to sequentially design experiments [6]. The same framework has been used in [7] for state tracking with active observation control where a Kalman-Like state estimator is developed and in [8] for designing primary user system power probes for actively learning interference channel gains. In this category, also constrained AL problems were formulated and solved by researchers in various fields which employed Constrained DP [9, 10]. The authors of [11] accomplished to actively identify human body states with biometric device sensing costs and the authors of [12] achieved to manage a sensor network with communication costs.

In this paper, we combine this Constrained DP framework with a sophisticated Bayesian Learning tool, the EP [13], for sequentially designing unlabelled data, basically feature vectors chosen from a continuous high dimensional space, and requesting their labels from an oracle. The purpose of this Constrained Bayesian AL (CBAL) design method is to learn a linear classifier with the minimum number of queries to the oracle while setting a limit on the expected number of class labels from a specific "unwanted" class. Moreover, an analytical implementation of the EP is proposed for the first time by utilizing recent advances in statistics from the econometrics research community [14]. The remainder of this paper is structured as follows: Section II provides the Bayesian Learning formulation and presents the EP. Section III elaborates on the optimal CBAL for a linear classifier. In Section IV, the simulation results obtained from the application of the proposed technique are shown and in Section V conclusions and future work in this topic are given.

2. BAYESIAN LEARNING AND EXPECTATION PROPAGATION

We begin by defining the normalized version of the unknown linear classifier that we wish to learn:

$$y = \begin{cases} +1 & \text{if } \mathbf{h}\,\mathbf{x}^{\intercal} \leq 1 \\ -1 & \text{if } \mathbf{h}\,\mathbf{x}^{\intercal} > 1 \end{cases} \tag{1}$$

where $\mathbf{x}$ is a feature vector in $\mathbb{R}^N$ and y is its corresponding label. In this section, we present a probabilistic way to learn the N-dimensional parameter vector, $\mathbf{h}$, of the linear classifier. The true value of $\mathbf{h}$ will be denoted as $\mathbf{h}^*$ from here on. To describe Bayesian Learning in detail, first we need to define the label conditional likelihood as the probability of

This work was supported by the European Research Council under the project "AGNOSTIC: Actively Enhanced Cognition based Framework for Design of Complex Systems".

y conditioned on the unknown parameter $\mathbf{h}^*$ and the feature vector $\mathbf{x}$:

$$\Pr[y|\mathbf{h}=\mathbf{h}^*,\mathbf{x}^\intercal]=\begin{cases}1 & \text{if } y=+1 \text{ and } \mathbf{h}\,\mathbf{x}^\intercal\leq 1\\ 0 & \text{if } y=+1 \text{ and } \mathbf{h}\,\mathbf{x}^\intercal> 1\\ 1 & \text{if } y=-1 \text{ and } \mathbf{h}\,\mathbf{x}^\intercal> 1\\ 0 & \text{if } y=-1 \text{ and } \mathbf{h}\,\mathbf{x}^\intercal\leq 1\end{cases}. \quad (2)$$

This expression is actually a threshold likelihood metric determined by the label observation, y, and the feature vector $\mathbf{x}$. According to the *version space duality* [15], when we deal with learning linear classifiers, feature vectors are hyperplanes in the parameter or version space and vice versa. Hence, when a learning procedure tries to estimate the parameters of a hyperplane, the version, it actually tries to localize a point in the parameter or version space. In addition, by combining a feature vector and its respective label, an inequality is obtained which in the $\mathbf{h}$ space, or version space, represents a linear inequality. Therefore, the likelihood function may also be thought of as a halfspace defined by $\mathbf{x}$ and y in the version space.

Now, let us assume that following t queries, or unlabelled feature vectors, $\mathbf{x}_{0:(t-1)}=\{\mathbf{x}(0),..,\mathbf{x}(t-1)\}$, the learner has obtained t responses, or labels, $y_{0:(t-1)}=\{y_0,..,y_{(t-1)}\}$, which all together constitute the data known until the $(t-1)$ training sample pair, D_{t-1}. After a new feature vector $\mathbf{x}(t)$ and its label, y_t, the $\mathbf{h}$ posterior probability density function (pdf) is expressed as:

$$f_{t+1}(\mathbf{h})=\frac{\Pr[y_t|\mathbf{h}=\mathbf{h}^*,\mathbf{x}(t)]\;f_t(\mathbf{h})}{\Pr[y_t|\mathbf{x}(t),D_{t-1}]} \quad (3)$$

which indicates the probability of where $\mathbf{h}^*$ lies in the $\mathbf{h}$ space given D_t. Here, we have exploited the fact that the label y_t is conditionally independent of the previous labels $y_{0:(t-1)}$ and feature vectors $\mathbf{x}_{0:(t-1)}$ given $\mathbf{h}=\mathbf{h}^*$ and $\mathbf{x}(t)$. Moreover, the denominator term is called the marginal likelihood and even though it is difficult to calculate, it is actually a normalization constant which guarantees that the posterior pdf integrates to 1. Alternatively, the posterior pdf expressed in (3) can be written in a non-recursive form as:

$$f_{t+1}(\mathbf{h})=\frac{\prod_{i=0}^{t}\Pr[y_i|\mathbf{h}=\mathbf{h}^*,\mathbf{x}(i)]}{\prod_{i=0}^{t}\Pr[y_i|\mathbf{x}(i),D_{i-1}]}\;f_0(\mathbf{h}) \quad (4)$$

where $f_0(\mathbf{h})$ is the prior pdf and again the denominator term is a normalization factor whose computation will be shown unnecessary.

2.1. The Expectation Propagation algorithm

Now, each likelihood function can be expressed as $l_i(\mathbf{h})=\Pr[y_i|\mathbf{h}=\mathbf{h}^*,\mathbf{x}(i)]$ and hence the likelihood function product of (4) is now $\prod_{i=0}^{t} l_i(\mathbf{h})$, which is basically a product of halfspace indicator functions. In this subsection, we show how to approximate $\prod_{i=0}^{t} l_i(\mathbf{h})$ and thus the deriving posterior pdf using EP [13]. The rationale of the EP is to approximate this product by finding an approximation $\tilde{l}_i(\mathbf{h})$ for each $l_i(\mathbf{h})$. This is done by initializing arbitrarily the likelihood function approximations and iteratively filtering each one of them considering the rest approximations stable. This filtration process is based on minimizing the Kullback-Leibler (KL) divergence of $l_j(\mathbf{h})\prod_{i=0,i\neq j}^{t} l_i(\mathbf{h})$ and $\tilde{l}_j(\mathbf{h})\prod_{i=0,i\neq j}^{t} l_i(\mathbf{h})$. A detailed algorithmic description of EP is presented in Algo. 1.

Algorithm 1 The Expectation Propagation algorithm

Initialize arbitrarily $\{\tilde{l}_0(\mathbf{h}),\tilde{l}_1(\mathbf{h}),...,\tilde{l}_t(\mathbf{h})\}$
for $k=1:N_{EP}$ **do**
  for $j=0:t$ **do**
    $\tilde{l}_j(\mathbf{h}):=$
    $$\arg\min_{\tilde{l}_j(\mathbf{h})} KL\left(l_j(\mathbf{h})\prod_{i=0,i\neq j}^{t}\tilde{l}_i(\mathbf{h})\parallel\tilde{l}_j(\mathbf{h})\prod_{i=0,i\neq j}^{t}\tilde{l}_i(\mathbf{h})\right)$$
  end for
end for

Next, we show how to tackle *analytically* the KL divergence minimization, the critical step of the EP algorithm, without relying on numerical quadratures or independence assumptions between the latent variables. This will lead to greater accuracy and faster implementation of this sophisticated tool. Here, each approximation in the EP algorithm is considered to be a multivariate normal (MVN) pdf. Consequently, the product of MVN pdf's, which appears in the KL divergence minimization step, based on Gaussian identities is also an MVN pdf. More specifically, if $\tilde{l}_i(\mathbf{h})=\mathcal{N}(\mathbf{h};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ for $i=0,...,t$, where $\boldsymbol{\mu}_i$ are the mean row vectors and $\boldsymbol{\Sigma}_i$ are the covariance matrices, then their product, $\prod_{i=0}^{t}\tilde{l}_i(\mathbf{h})$, is an un-normalized MVN pdf proportional to an MVN pdf, $\mathcal{N}(\mathbf{h};\boldsymbol{\mu}_{tot},\boldsymbol{\Sigma}_{tot})$, where $\boldsymbol{\Sigma}_{tot}^{-1}=\sum_{i=0}^{t}\boldsymbol{\Sigma}_i^{-1}$, $\boldsymbol{\mu}_{tot}=\left(\sum_{i=0}^{t}\boldsymbol{\mu}_i\boldsymbol{\Sigma}_i^{-1}\right)\boldsymbol{\Sigma}_{tot}$ and all vectors are assumed to be row vectors.

Hence, the second part of the KL divergence in the core stage of the EP method, $\tilde{l}_j(\mathbf{h})\prod_{i=0,i\neq j}^{t}\tilde{l}_i(\mathbf{h})$, and the approximation product in the first part, $\prod_{i=0,i\neq j}^{t}\tilde{l}_i(\mathbf{h})$, are basically un-normalized MVN pdf's. For notation simplification, $\prod_{i=0,i\neq j}^{t}\tilde{l}_i(\mathbf{h})$ will be symbolized from now on as $\tilde{l}_{-j}(\mathbf{h})$. Now, as far as the KL divergence minimization is concerned, when approximations within the exponential family are used, then this is achieved by *moment matching* [16]. *Moment matching* means that the two functions whose KL divergence

needs to be minimized must have the same moments and since the second function is an un-normalized MVN one, this results to matching the 0_{th}, 1_{st} and 2_{nd} moments of the two parts. This basically indicates that the function to be refined in each EP step, $\tilde{l}_j(\mathbf{h})$, must be adjusted so that the moments of $\tilde{l}_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$ are equal to the ones of $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$.

Now, let us examine the function $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$. First, we have already shown that $\tilde{l}_{-j}(\mathbf{h})$ is an un-normalized MVN function and we have described $l_j(\mathbf{h})$ as a halfspace indicator function. Thus, $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$ is actually a one-side truncated multivariate Gaussian and what we need is to calculate its 0_{th}, 1_{st} and 2_{nd} moments, q, $\mathbf{q}$ and $\mathbf{Q}$. This can be found in detail in Appendix A of [17] where we utilized the moment formulas for doubly truncated MVNs [14] after reformulating them. Once these moments are computed, $\tilde{l}_j(\mathbf{h})$ is defined as an MVN pdf with covariance matrix $\boldsymbol{\Sigma}_j^{-1} = \mathbf{Q}^{-1} - \boldsymbol{\Sigma}_{-j}^{-1}$ and mean $\boldsymbol{\mu}_j = \left(\mathbf{q}\ \mathbf{Q}^{-1} - \boldsymbol{\mu}_{-j}\ \boldsymbol{\Sigma}_{-j}^{-1}\right)\boldsymbol{\Sigma}_j$. We also need to highlight that matching the 0_{th} moments does not offer essentially better approximations, because multiplying $\tilde{l}_j(\mathbf{h})$ with a constant may lead to unwanted results in this iterative filtration process. Still, we mentioned this earlier as part of the moment matching process for the sake of completeness.

3. CONSTRAINED BAYESIAN ACTIVE LEARNING OF A LINEAR CLASSIFIER

The goal of this paper is to sequentially design feature vectors, $\mathbf{x}$, whose labels are revealed to the learner one by one in order to learn as fast as possible the parameter vector, $\mathbf{h}^*$, and while ensuring that the number of feature vectors of class $y=-1$ for a specific query budget is always below a certain limit. This means that assuming a limited number of N_T queries and thus feature vectors, $\{\mathbf{x}(0),...,\mathbf{x}(N_T-1)\}$ and their corresponding labels, $\{y_0,...,y_{N_T-1}\}$, we wish to minimize the uncertainty of our knowledge about $\mathbf{h}^*$, formally represented by the entropy of $f_{N_T}(\mathbf{h})$, subject to maintaining the sum of $y_t=-1$, where $t=0,...,N_T-1$, below a threshold and which is equivalent to controlling the sum of $y_t=+1$, where $t=0,...,N_T-1$, above a corresponding limit. In the previous section, we showed the recursive Bayesian update (3) which modifies our knowledge about $\mathbf{h}^*$ step by step. This will be our main tool for handling the iterative nature of this proactive feature vector design strategy.

3.1. The DP formulation of the Constrained Bayesian AL problem

Next, we investigate the optimal training sample design policy that should be chosen in each step of this recursive Bayesian estimation process in order to optimally reduce the posterior pdf entropy after N_T feature vectors, $\{\mathbf{x}(0),...,\mathbf{x}(N_T-1)\}$, with their corresponding labels, $y_{0:(N_T-1)}$, subject to $\frac{1}{N_T}\sum_{t=0}^{N_T-1}\mathbf{1}_{\{y_t=-1\}} \leq \alpha'$ or $\frac{1}{N_T}\sum_{t=0}^{N_T-1}\mathbf{1}_{\{y_t=+1\}} \geq \alpha$, where $\mathbf{1}_{\{..\}}$ is the indicator function, α' is the threshold ratio of class $y=-1$ occurrences, α is the lower limit ratio of class $y=+1$ occurrences and $\alpha=1-\alpha'$. From here on, we employ α for the formulation of our problem. The constraint can also be written as $\sum_{t=0}^{N_T-1} y_t \geq (2\alpha-1)N_T$. This multistage constrained optimization problem can be expressed in the spirit of DP [18] as finding the optimal query rule that maps $\{f_0,..,f_{N_T-1}\}$ to $\{\mathbf{x}(0),..,\mathbf{x}(N_T-1)\}$ in order to achieve the maximum average entropy reduction from the $f_0(\mathbf{h})$ to the $f_{N_T}(\mathbf{h})$ pdf subject to the aforementioned constraint. In a formal manner, we seek the optimal query/feature vector design policy $\pi^*_{0:(N_T-1)} = \{\mathbf{x}(0)=\mu^*(f_0),..,\mathbf{x}(N_T-1)=\mu^*(f_{N_T-1})\}$ which solves the following constrained optimization problem over all possible label sequences derived by this query policy:

$$\max_{\pi} \quad E^{\pi}[\mathcal{H}(f_0)-\mathcal{H}(f_{N_T})|\mathbf{x}(N_T-1),D_{N_T-2}] \tag{5a}$$

$$\text{s.t.} \quad E^{\pi}\left[\sum_{t=0}^{N_T-1} y_t|\mathbf{x}(N_T-1),D_{N_T-2}\right] \geq (2\alpha-1)N_T \tag{5b}$$

where $\mathcal{H}$ is the entropy operator of a pdf and $E^{\pi}[.]$ is the operator of the average value over all possible label sequences derived by an abstract policy π.

Before we continue though with the DP solution of our constrained multistage problem, let us first redefine the multivariate cumulative distribution function (cdf) in a more "natural" than the usual way. Assuming a hyperplane, $\mathbf{x}\ \mathbf{w}^{\intercal}=1$, we alternatively determine the cdf C of a multivariate pdf f as:

$$C(\mathbf{w}) = \Pr[\mathbf{x}\ \mathbf{w}^{\intercal} \leq 1] = \int\limits_{\mathbf{x}\ \mathbf{w}^{\intercal}\leq 1} f(\mathbf{x})\ dV_{\mathbf{x}}. \tag{6}$$

In our case, this means that the posterior cdf after the $(t-1)$ step, $C_t(\mathbf{x})$, is expressed as:

$$C_t(\mathbf{x}) = \Pr[\mathbf{h}\ \mathbf{x}^{\intercal} \leq 1|\mathbf{h}=\mathbf{h}^*, D_{t-1}] = \int\limits_{\mathbf{h}\ \mathbf{x}^{\intercal}\leq 1} f_t(\mathbf{h})\ dV_{\mathbf{h}}. \tag{7}$$

Using this cdf definition, the optimal feature vector design policy of the Constrained DP problem (5) is found to be $\pi^*_{0:(N_T-1)} = \{\mathbf{x}(0) = C_0^{-1}(\alpha_0),..,\mathbf{x}(N_T-1) = C_{N_T-1}^{-1}(\alpha_{N_T-1})\}$ where $\alpha_0=\alpha$, $\alpha_k = \frac{2\alpha N_T - k - \sum_{t=0}^{k-1} y_t}{2(N_T-k)}$ for $k=1,...,(N_T-1)$ and $\mathbf{x}(k)=C_k^{-1}(\alpha_k)$ is equivalent to $C_k(\mathbf{x}(k))=\alpha_k$. A detailed derivation of this solution can be found in [17].

3.2. The Necessity of Exploration

Reducing the uncertainty of our knowledge about $\mathbf{h}^*$ must be performed by approaching this exact value uniformly from

all directions. This means that the training samples in an AL process must be diversified and this can be accomplished by choosing hyperplanes in the version space of random direction uniformly which promotes *exploration*. Therefore, we need first to define how to uniformly sample a random direction $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is a unit vector. This problem is related to the uniform unit hypersphere point picking which has been thoroughly described in [3, 8]. Hence, in order to produce a feature vector which represents a hyperplane of random direction, $\mathbf{x}(t)$ must be parallel to a randomly generated $\boldsymbol{\theta}$, $\mathbf{x}(t) = \beta\boldsymbol{\theta}$ where $\beta \in \mathbb{R}$, and it must also satisfy $C_t(\mathbf{x}(t)) = \alpha_t$ according to our previous analysis. In a formal manner, this is expressed using (6) as:

$$\int\limits_{\mathbf{h}\ \beta\boldsymbol{\theta}^\intercal \leq 1} f_t(\mathbf{h})\, dV_{\mathbf{h}} = C_t(\beta\boldsymbol{\theta}) = \alpha_t. \tag{8}$$

At this point, we make use of the Gaussian approximation of each step's posterior pdf which we developed in Section II with the help of EP. In accordance with that result, $f_t(\mathbf{h})$ can be approximated by the normalized version of $\prod_{i=0}^{t-1} \tilde{l}_i(\mathbf{h})$ which we denote as $\tilde{f}_t(\mathbf{h})$. So, (8) now becomes:

$$\int\limits_{\mathbf{h}\ \beta\boldsymbol{\theta}^\intercal \leq 1} \tilde{f}_t(\mathbf{h})\, dV_{\mathbf{h}} = \alpha_t \tag{9}$$

After some processing, we obtain that $\beta = \frac{1}{F^{-1}(\alpha_t; c_1, c_2)}$ where $F^{-1}(.)$ is the inverse cdf of the univariate normal pdf with mean c_1 and variance c_2. Furthermore, $c_1 = \boldsymbol{\theta}\ \tilde{\boldsymbol{\mu}}^\intercal(t)$, where $\tilde{\boldsymbol{\mu}}(t)$ is the mean row vector of $\tilde{f}_t(\mathbf{h})$, and $c_2 = \sum_{i=1}^{N} \theta_i \boldsymbol{\theta}(\tilde{\boldsymbol{\Sigma}}_{:,i}(t))^\intercal$, where $\tilde{\boldsymbol{\Sigma}}_{:,i}(t)$ is the i_{th} column of the covariance matrix of $\tilde{f}_t(\mathbf{h})$.

4. RESULTS

In this section, we show the $\mathbf{h}^*$ estimation error achieved by the proposed CBAL method depending on the number of training samples. The $\mathbf{h}^*$ estimation error metric at each time step is defined as the normalized root-square error $\frac{\|\hat{\mathbf{h}}(t) - \mathbf{h}^*\|}{\|\mathbf{h}^*\|}$ and basically demonstrates the learning efficiency of our method. The estimated parameter vector at each step, $\hat{\mathbf{h}}(t)$, is considered as the $\tilde{\boldsymbol{\mu}}(t)$ of the EP. The error figure results are obtained as the average of the error metric defined earlier over 100 random draws of parameter vectors $\mathbf{h}^*$. The examined scenarios consider $N = 5$ with $\alpha = \{\ 0.7,\ 0.9\}$ and $N = 10$ with $\alpha = 0.7$ and the "budget" of queries to the oracle is considered to be $N_T = 200$. Additionally, for the EP implementation, the outer loop iterations, N_{EP}, are chosen to be 5 and the prior pdf is a MVN of sufficiently vast circular covariance matrix.

Next, in Fig. 1 we examine the performance of the CBAL technique for all the aforementioned scenarios. First, we observe that the number of required training samples to reach a specific error percentage increases as the problem dimensions, N, grow. Second, as α increases, the necessary training samples to achieve again a particular estimation error also increase. Furthermore, a metric α_{sim} is provided which examines the simulated "wanted" class label occurrences, $\alpha_{sim} = \frac{\sum_{t=0}^{N_T-1} \mathbf{1}_{\{y_t=+1\}}}{N_T}$. As far as the resulting α_{sim} metric is concerned, its values are 0.68, 0.87 and 0.67 for $\alpha = 0.7$ and $N = 5$, for $\alpha = 0.9$ and $N = 5$ and for $\alpha = 0.7$ and $N = 10$ respectively.

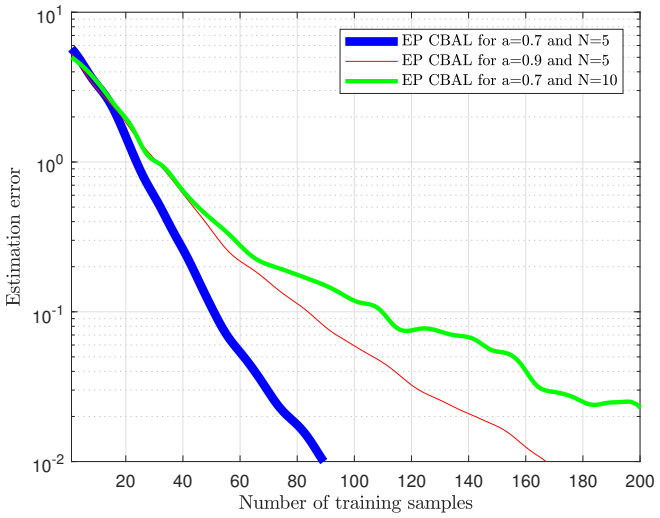

Fig. 1. Classifier estimation error vs number of training samples using the CBAL method

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an AL method in order to learn a linear classifier as fast as possible and while limiting the expected ratio of labels from one "unwanted" class under a specific threshold. This problem was formulated within the Constrained DP framework and its optimal feature vector design policy was implemented with the help of an advanced, fast and accurate Bayesian Learning technique, the EP. The performance of this method was demonstrated through numerical simulations and we confirmed that the simulated metric for the label ratio of the "wanted" class, α_{sim}, is satisfactorily close to the target, or design, ratio α. As part of our future work, the learning convergence rate of the proposed CBAL method will be studied, since theoretical guarantees for the number of iterations needed to approach the learning solution within some error bound are an essential part of the AL setting.

6. REFERENCES

[1] D. A. Cohn, A. Ghahramani, and M. I. Jordan, "Active Learning with Statistical Models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, 1996.

[2] B. Gopalakrishnan and N. D. Sidiropoulos, "Cognitive Transmit Beamforming from Binary CSIT," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 895–906, Feb. 2014.

[3] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Centralized Power Control in Cognitive Radio Networks Using Modulation and Coding Classification Feedback," *IEEE Trans. Cognitive Commun. and Networking*, vol. 2, no. 3, pp. 223–237, Sept. 2016.

[4] U. Louche and L. Ralaivola, "From Cutting Planes Algorithms to Compression Schemes and Active Learning," in *Proc. International Joint Conf. on Neural Networks (IJCNN)*, Jul. 2015.

[5] K. Chaloner and I. Verdinelli, "Bayesian Experimental Design: A Review," *Stat. Sci.*, vol. 10, no. 3, pp. 273–304, Aug. 1995.

[6] I. Ben-Gal and M. Caramanis, "Sequential DOE via dynamic programming," *IIE Trans.*, vol. 34, no. 12, pp. 1087–1100, May 2002.

[7] D.S. Zois, M. Levorato, and U. Mitra, "Active Classification for POMDPs: A Kalman-Like State Estimator," *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6209–6214, Dec. 2014.

[8] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Interference Constraint Active Learning with Uncertain Feedback for Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4654–4668, Jul. 2017.

[9] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration," *Springer US Autonomous Robots*, vol. 39, no. 4, pp. 555–571, Dec. 2015.

[10] L.M. Lopez-Ramos, A. G. Marques, and J. Ramos, "Jointly Optimal Sensing and Resource Allocation for Multiuser Interweave Cognitive Radios," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5954–5967, Nov. 2014.

[11] D.S. Zois and U. Mitra, "Active State Tracking With Sensing Costs: Analysis of Two-States and Methods for n-States," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2828–2843, Jun. 2017.

[12] J.L. Williams, J.W. Fisher, and A.S. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, Aug. 2007.

[13] T. Minka, *A family of algorithms for approximate Bayesian inference*, Ph.D. thesis, MIT, Cambridge, MA, 2001.

[14] R. Kan and C. Robotti, "On Moments of Folded and Truncated Multivariate Normal Distributions," *Journal of Computational and Graphical Statistics*, Apr. 2017.

[15] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 2000.

[16] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA, USA, 2012.

[17] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Constrained Bayesian Active Learning of Interference Channels in Cognitive Radio Networks," *https://arxiv.org/abs/1710.08335*, 2017.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, 1995.