# LEARNING TO SATISFY

*Frederic Thouin*[†]*, Mark Coates*[†]*, Brian Erikkson*[⋆]*, Robert Nowak*[⋆]*, Clayton Scott*[‡]

[†]Department of Electrical Computer Engineering, McGill University
[⋆]Department of Electrical Computer Engineering, University of Wisconsin
[‡]Department of Electrical Engineering and Computer Science, University of Michigan

**ABSTRACT**

This paper investigates a class of learning problems called *learning satisfiability* (LSAT) problems, where the goal is to learn a set in the input (feature) space that satisfies a number of desired output (label/response) constraints. LSAT problems naturally arise in many applications in which one is interested in the class of inputs that produce desirable outputs, rather than simply a single optimum. A distinctive aspect of LSAT problems is that the output behavior is assessed only on the solution set, whereas in most statistical learning problems output behavior is evaluated over the entire input space. We present a novel support vector machine (SVM) algorithm for solving LSAT problems and apply it to a synthetic data set to illustrate the impact of the LSAT formulation.

***Index Terms—*** Machine Learning, Satisfiability, SVM, Minimum Volume Sets, One-class Neighbor Machines.

## 1. INTRODUCTION

In most statistical learning problems, one is interested in minimizing a risk function such as expected squared error or probability of error. However, in many applications, one is interested in a solution to the learning problem that satisfies several criteria simultaneously, rather than simply optimizing one. In this paper, we introduce and study *learning satisfiability* (LSAT) problems, a class of learning problems where the goal is to learn a set in the input (feature) space that satisfies a number of desired properties expressed in terms of expectations and/or event probabilities.

LSAT problems arise in a number of important applications in classification and statistics. An example is an extension of the false discovery rate approach for controlling the number of false positives in multiple hypothesis tests [1]. Another example is the portfolio selection problem where one is interested in identifying a set of stocks based on historical data such that not only is the expected return positive but large losses are rare. In this latter problem, we seek the largest set in the input space such that: (i) the expected output value at every point in the set is non-negative; and (ii) the probability that the output stays above a lower limit is guaranteed to be large. Both criteria are in the form of constraints and express different measures of *confidence* in a favorable output.

The rest of the paper is organized as follows. In Sect. 2, we mathematically define LSAT problems, and we then present a methodology and algorithm to solve them in Sect. 3. In Sect. 4, we compare our algorithm to standard weighted support vector machine (SVM) techniques on a synthetic data set. We conclude and propose future research avenues in Sect. 5.

### 1.1. Related Work

One example of learning with multiple criteria is the Neyman-Pearson (NP) learning problem, in which one seeks a classifier that minimizes the false negative rate subject to a constraint on the false positive rate [2, 3]. An important distinction between NP learning and LSAT problems is that in LSAT problems output behavior is assessed on the solution set, whereas in NP learning (as well as most other standard learning problems) one is concerned with output behavior over the entire input space. Thus, LSAT criteria generally involve conditional probabilities/expectations that are functions of the target set, i.e. conditioning is on membership in the *output* set. In contrast, the conditioning in the constraints used in Neyman-Pearson learning (and in the performance metrics used in many standard classification approaches) is on the *input* class label. This difference leads to requirements for new theory and learning methods.

LSAT problems are also related to classical satisfiability (SAT) problems, most closely perhaps to stochastic SAT (SSAT) problems [4, 5]. SSAT problems involve criteria that depend on a mixture of controllable decision variables and stochastic variables, and the main objective is to determine whether there exist values for the decision variables such that the probability that the criteria are satisfied exceeds a certain threshold. A major difference between SSAT and LSAT problems is that the randomness in SSAT problems is typically known and therefore learning from data is not involved. Also, LSAT does not involve decision variables, but focuses on identification of the (possibly empty) set of inputs that satisfy stochastic criteria. Finally, since LSAT involves the maximization of set size, there are relationships with one-class neighbor (and support vector) machines and methods for learning minimum volume sets [6, 7, 8].

## 2. LSAT PROBLEM FORMULATION

To formally define our problem, let us first introduce the following notation. Features $X$ are elements in the input space $\mathcal{X}$. An output $Y \in \mathcal{Y}$ is associated with each input. Let $\mathcal{P}$ denote a collection of probability measures on $\mathcal{X} \times \mathcal{Y}$. Each pair $(X, Y)$ is distributed independently and identically according to an unknown probability measure $P \in \mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$. We are interested in identifying the largest set in the input space where certain output constraints are met. Let $\mathcal{G}$ denote a collection of candidate sets and let $C : \mathcal{G} \times \mathcal{P} \rightarrow R^{k+1}$ be a constraint function mapping each set and probability measure to a $(k+1)$-dimensional vector of real numbers. For a given probability measure $P$, we are interested in the largest set $G \in \mathcal{G}$ that satisfies the constraint $C(G, P) \geq 0$, where the inequality is applied element-by-element. Let $\mu(G)$ denote a positive measure of choice, then

$$\max_{G \in \mathcal{G}} \mu(G) \text{ subject to } C(G, P) \geq 0$$

A solution may not exist, depending on the nature of the constraints and $P$ (in such cases, we consider the empty set to be a default solution). An alternate expression of the LSAT problem, which also lends itself naturally to the identification of the $\mu$-largest feasible set, is to express one of the constraint criteria as a risk function to be minimized subject to the other constraints. Let $R(G, P)$ be a risk function chosen such that it is minimized by the largest set satisfying $C_0(G, P) \geq 0$.

$$\min_{G \in \mathcal{G}} R(G, P) \text{ subject to } C_j(G, P) \geq 0,\ j = 1, \ldots, k$$

We wish to stress that any such risk function must satisfy two important properties with respect to the other constraints: (i) if there exists a non-empty solution to the standard LSAT formulation, the (constrained) risk minimizer must coincide with this solution, and (ii) if there is no solution, the empty set must have smaller risk than any set failing to satisfy $C_0$.

### 2.1. Two Types of Constraints

One of the distinctive features of LSAT problems is that the output behavior is assessed only on the solution set, whereas in most statistical learning problems output behavior is evaluated over the entire input space. We consider two types of set-based output constraints.

1. **Point-wise Constraint:** $C(G, P) = C(x, G, P)$ is a function of the input variable $x$, and the constraint takes the form $C(x, G, P) \geq 0,\ \forall\, x \in G$.
2. **Set-average Constraint:** $C(G, P)$ is only a function of the set $G$, and the constraint $C(G, P) \geq 0$ is only satisfied "on-average" over the set $G$.

Examples of the point-wise type of constraint include $E[Y|X = x] \geq 0$ and $P(Y \geq L|X = x) - p \geq 0, \forall x \in G$. Corresponding examples for the set-average constraint type are $E[Y|X \in G] \geq 0$ and $P(Y \geq L|X \in G) - p \geq 0$.

## 3. SOLUTIONS TO LSAT PROBLEMS

### 3.1. Methodology

We are interested in identifying the set $G \in \mathcal{G}$ that satisfies the constraints $C(G, P) \geq 0$ and has minimum risk $R(G, P)$. However, since the probability measure $P$ is unknown, we aim to learn this set from a training sample $\{X_i, Y_i\}_{i=1}^n$. Suppose that we form empirical versions of the constraint functions $C_i(G, \widehat{P})$ and risk $R(G, \widehat{P})$, based on the empirical distribution $\widehat{P}$ of the training sample. For the remainder of the paper we will no longer explicitly indicate the dependence of the constraints on the underlying probability measure $P$, simply writing $C(G) = C(G, P)$, $\widehat{C}(G) = C(G, \widehat{P})$, $R(G) = R(G, P)$, and $\widehat{R}(G) = R(G, \widehat{P})$. Define the optimal set

$$G^* = \arg\min_{G \in \mathcal{G}} R(G) \text{ subject to } C_j(G) \geq 0,\ j = 1, \ldots, k.$$

Let $\epsilon_0, \ldots, \epsilon_k > 0$ be fixed and define

$$\widehat{G} = \arg\min_{G \in \mathcal{G}} \widehat{R}(G) \text{ subject to } \widehat{C}_j(G) \geq -\epsilon_1,\ j = 1, \ldots, k.$$

By allowing constraints to be violated by the small tolerances $\epsilon_i$, we are can relate the performance of $\widehat{G}$ to that of $G^*$.

**Lemma 1.** *If* $\sup_{G \in \mathcal{G}} |R(G) - \widehat{R}(G)| \leq \epsilon_0$ *and* $\sup_{G \in \mathcal{G}} |C_j(G) - \widehat{C}_j(G)| \leq \epsilon_j$ *for* $j = 1, \ldots, k$ *then*

$$R(\widehat{G}) \leq R(G^*) + 2\epsilon_0 \quad \textit{and} \quad C_j(\widehat{G}) \geq -2\epsilon_j,\ \ j = 1, \ldots, k$$

*Proof.* Under the assumed deviation bounds $\widehat{C}_j(G^*) \geq C_j(G^*) - \epsilon_j \geq -\epsilon_j$, which implies that $G^*$ is in the empirical constraint set. Thus $\widehat{G}$ minimizes $\widehat{R}$ subject to the empirical constraints: $R(\widehat{G}) \leq \widehat{R}(\widehat{G}) + \epsilon_0 \leq \widehat{R}(G^*) + \epsilon_0$. Applying the deviation bound to $\widehat{R}(G^*)$ produces the result. □

### 3.2. Support Vector Machine (SVM) algorithm

We describe an SVM algorithm for solving the empirical constrained optimization problem in some common LSAT scenarios. We focus on the case where there is one pointwise constraint $C_0$, and we assume that it is possible to identify an associated risk $R_0$ (and empirical version $\widehat{R}_0$) that satisfies the properties identified in Sect. 2.

Our algorithmic approach is to map the constrained optimization into a cost-sensitive classification problem. We associate with each data point a cost of inclusion and of exclusion. This cost is a Lagrangian sum of the risk $\widehat{R}_0$ and individual cost terms for each setwise constraint. We thus map each data point $(X_i, Y_i)$ to a triple $(X_i, Z_i, \gamma_i)$, where $Z_i$ is a class label and $\gamma_i$ is the penalty incurred through misclassification of this point. In the LSAT setting, we have multiple constraints, so we generate a label $Z_{i,j}$ and a cost $\gamma_{i,j}$ for each data point $i$ and constraint $j$. In order to apply cost-sensitive classification, we must collapse the $Z_{i,j}$ to a single

label $Z_i$. For each constraint, we assign a weight $\lambda_j$ that provides a mechanism for adjusting the relative importance of each of them. Note that $\gamma_{i,0}$ is determined by the contribution of point $i$ to $\widehat{R}_0$ and $\lambda_0 = 1$. If $Z_{i,j} = 1$ for all $j$, then we set $Z_i = 1$ and $\gamma_i = \sum_{j=0}^{k} \lambda_j \gamma_{i,j}$. A similar procedure applies if $Z_{i,j} = 0$ for all $j$. The situation is more complicated if $Z_{i,j}$ differs for various constraints. In this case, we set $Z_i = 1$ and $\gamma_i = \sum_{Z_{i,j}=1} \lambda_j \gamma_{i,j}$. However, we also construct an auxiliary data point $(X_{\tilde{i}}, Z_{\tilde{i}}, \gamma_{\tilde{i}})$, with $Z_{\tilde{i}} = 0$ and $\gamma_{\tilde{i}} = \sum_{Z_{i,j}=0} \lambda_j \gamma_{i,j}$.

To solve the cost-sensitive classification, we iteratively apply a modified version of the cost-sensitive $2\nu$-SVM [9, 10]. Performance is dependent on the choice of kernel, as with any SVM, but we do address that issue here; the results we present are derived using a Gaussian kernel, and exploring a set of logarithmically-spaced variances. The $2\nu$-SVM solves the optimization problem in (1), where $w$ and $b$ determine the separating hyperplane in the kernel-space, $\varepsilon$ and $\rho$ are slack variables, $\nu_+$ and $\nu_-$ provide a method for globally adjusting the weight associated with exclusion and inclusion (to encourage large sets), and $n_+$ and $n_-$ are the number of points included in, and excluded from, the identified set.

$$\min_{w,b,\varepsilon,\rho} \frac{||w||^2}{2} - 2\nu_+\nu_-\rho + \frac{\nu_-}{n_+}\sum_{i\in I^+}\varepsilon_i\gamma_i + \frac{\nu_+}{n_-}\sum_{i\in I^-}\varepsilon_i\gamma_i \quad (1)$$

$$\begin{aligned} \text{s.t.} \quad Z_i(k(w,x_i)+b) &\geq \rho - \varepsilon_i \quad \text{for } i = 1,\ldots,n \\ \varepsilon_i &\geq 0 \quad \text{for } i = 1,\ldots,n \\ \rho &\geq 0. \end{aligned}$$

We conduct this optimization for a range of $\lambda_j$, $\nu_+$ and $\nu_-$, effectively conducting a grid search over these parameters to find the largest set. In each case, we test whether the set satisfies constraints by performing k-fold cross-validation on the input data. This algorithm is limited to constraints where one can identify an appropriate mapping to labels and costs. This can, however, be achieved for a wide range of important constraints, including those involving bounds on pointwise or set-average expectation or tail probabilities.

### 3.3. Example

Consider the portfolio selection problem outlined in the introduction. We are interested in the set $G \in \mathcal{X}$ of largest $P$-measure that satisfies $E[Y|X = x] \geq U$, for all $x \in G$, and $P(Y > L|X \in G) \geq p$. The parameters $U$, $L < U$, and $p > 0$ are specified by the user.

$$C(G,P) = \begin{bmatrix} \min_{x\in G} E[Y|X=x] - U \\ P(Y > L|X \in G) - p \end{bmatrix}$$

As discussed in [10], minimizing the risk $\widehat{R}_0(G) = \sum_{i=1}^{n}(U - y_i)(1_{x_i\in G} - 1_{x_i\in\overline{G}})$ can be achieved by assigning to each training point a class-label $Z_{i,0} = 1_{Y_i>U}$ and a cost $\gamma_{i,0} = |Y_i - U|$ and then applying a cost-sensitive classification algorithm. We now consider the case of the empirical set-average constraint $\widehat{C}_1 : \sum_{i=1}^{n} 1_{Y<L, X\in G} / \sum_{i=1}^{n} 1_{X\in G} < 1 - p$. This poses a greater challenge due to the inherent self-normalization in the constraint. Developing a correct mapping to labels and costs for this constraint is difficult, so we first consider an alternative constraint $\widehat{C}_1' : \sum_{i=1}^{n} 1_{Y<L, X\in G} < P^*(1-p)$ for a selected constant $P^*$. This constraint is now a bound on the joint probability of membership in G and $Y < L$, rather than on the conditional probability as in the original constraint. It is easier to identify a suitable mapping for this constraint. This can be achieved by assigning a label $Z_{i,1} = 1_{Y_i>L}$ and a cost $\gamma_{i,1} = 1$. The important observation is that constraints $\widehat{C}_1$ and $\widehat{C}_1'$ coincide when $P^* = 1_{X\in G_1^*}$, where we denote by $G_1^*$ the maximum probability set that satisfies constraint $C_1$.

The second step involves the combination of the labels and costs associated with $C_0$ and $C_1$. Using the procedure outlined above, this leads to the following set of costs. If $Y_i \geq U$, then $Z_i = 1$ and $\gamma_i = r_i^+$, whereas if $Y_i \leq L$, then $Z_i = 0$ and $\gamma_i = r_i^-$. If $L < Y_i < U$, then $Z_i = 1$, but we also construct a point $(X_{\tilde{i}}, Z_{\tilde{i}}, \gamma_{\tilde{i}})$ with $X_{\tilde{i}} = X_i$ and $Z_{\tilde{i}} = 0$.

$$r_i^+ = |y_i - U|1_{y_i>U} + \lambda_1 \quad (2)$$

$$r_i^- = |y_i - U| + \lambda_1 1_{y_i<L} \quad (3)$$

With this set of points, class labels and costs, we iteratively apply $2\nu$-SVM, jointly maximizing over $P^*$, $\nu^+$, $\nu^-$ and $\lambda_1$ to identify the largest probability set that minimizes the empirical risk $\widehat{R}_0(G)$ subject to satisfying the empirical constraints $\sum_{i=1}^{n} 1_{X\in G} \geq P^*$ and $\sum_{i=1}^{n} 1_{Y<L, X\in G} < P^*(1-p)$. Note that it is not necessary to explicitly maximize over $P^*$ in (1), because maximizing over $\nu_+$ and $\nu_-$ achieves this maximization implicitly.

**Table 1**. Values for $\lambda_1, \nu_+, \nu_-, \sigma$ (kernel parameter)

| Variable name | Values |
|---|---|
| $\lambda_1$ | $[0.01, 0.1, 1, 10]$ |
| $\nu_+, \nu_-$ | [0.1, 0.28, 0.46, 0.64, 0.82, 1.00] |
| $\sigma$ | $[10^{-4}, 10^{-2.4}, 10^{-0.8}, 10^{0.8}, 10^{2.4}, 10^{4}]$ |

## 4. EXPERIMENTS

To test our approach and illustrate the impact of the LSAT formulation, we attempt to solve the problem introduced in Sect. 3.3 for a synthetically generated data set. The data set is composed of three easily identifiable clouds of points. All the points in the top-right cloud (Fig. 2) have $y > U$, all points in the bottom-left cloud have $y < L$ and the top-left cloud mainly includes point with $y > U$, but also enough points with $y < L$ such that $C_1$ is violated for this cluster of points. We compare our approach (LSAT) to a regular weighted SVM (WSVM) approach that only tries to identify the largest U-level set. Note that this WSVM solves the same optimization problem as (1), but the risk terms in (2) and (3) do not include

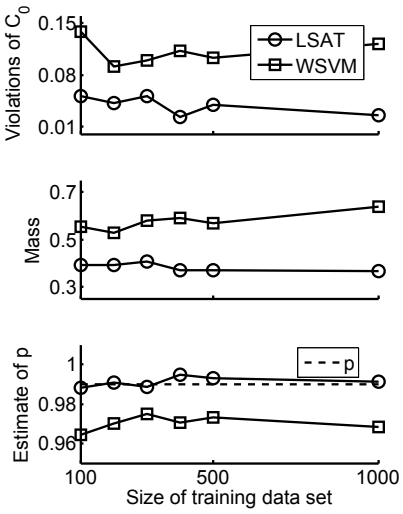


**Fig. 1**. Synthetic data set. Fraction of included points violating $C_0$ (TOP), mass (MIDDLE) and probability of avoiding small return (BOTTOM) of $G$ as a function of number of points used for training. Testing was performed using 2000 data points.

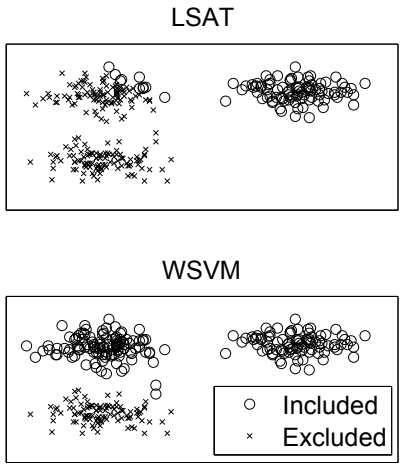


**Fig. 2**. Points included by LSAT and WSVM (synthetic data set).

the $\lambda_1$ terms, and there is no cross-validation to check that the empirical constraint $\widehat{C}_1$ is satisfied. In both cases, we perform a grid search on the parameters shown in Tab. 1 to obtain the best solution ($\lambda_1$ only applies to LSAT).

We compare the performance of the algorithm for different training set sizes $N_{train}$ ($[100, 200, 300, 400, 500, 1000]$). For each value of $N_{train}$, we average results over five training sets generated by randomly selecting points out of the training data set of size 1500 and reserve 2000 points for testing. We plot the fraction of points violating $C_0$, the mass of the selected sets as a fraction of the entire set and an estimate $\widehat{p}$. From Fig. 1, we can see that the LSAT approach is successful in satisfying the set-average constraint $C_1$ at the expense of generating a level-set with smaller mass. On the other hand, the standard WSVM includes more points in sets, hence the higher mass, but fails to satisfy $C_1$ and also has a higher fraction of points violating the point-wise constraint $C_0$. The trade-off is shown explicitly in Fig. 2, WSVM includes points from the top-left cloud whereas LSAT excludes them in order to satisfy $C_1$.

## 5. CONCLUSIONS

This paper introduced a new learning framework for handling LSAT problems and an algorithm based on weighted SVM to solve them. Using a simple synthetic data set, we showed the trade-off between the competing constraints of risk and return; by reducing risk, the LSAT approach selected a set $G$ with smaller mass than the weighted SVM approach. The future work will be directed at testing our approach and algorithm on real-life data as well as developing a bilevel optimization framework to replace the current grid search on the various optimization parameters.

## 6. REFERENCES


[1] J.D. Storey, "A direct approach to false discovery rates," *J.R. Statist. Soc. (Series B)*, vol. 64, no. 3, pp. 479–498, Mar. 2002.

[2] A. Cannon, J. Howse, D. Hush, and C. Scovel, "Learning with the Neyman-Pearson and min-max criteria," Tech. Rep. LA-UR 02-2951, Los Alamos National Laboratory, 2002.

[3] C. Scott and R. Nowak, "A Neyman-Pearson approach to statistical learning," *IEEE Trans. Information Theory*, vol. 51, no. 11, pp. 3806–3819, Nov. 2005.

[4] M. Majercik and M. L. Littman, "Contingent planning under uncertainty via stochastic satisfiability," *Artificial Intell.*, vol. 147, no. 1-2, pp. 119–162, Feb. 2003.

[5] S. Manandhar, A. Tarim, and T. Walsh, "Scenario-based stochastic constraint programming," in *Proc. Int. J. Conf. Artificial Intelligence (IJCAI)*, Acapulco, Mexico, Aug. 2003.

[6] A. Muñoz and J.M. Moguerza, "Estimation of high-density regions using one-class neighbor machines," *IEEE Trans. Patt. Analy. and Mach. Intell.*, vol. 28, no. 3, pp. 476–480, Mar. 2006.

[7] C. Scott and R. Nowak, "Learning minimum volume sets," *J. Machine Learn. Res.*, vol. 7, pp. 665–704, Apr. 2006.

[8] W. Polonik, "Minimum volume sets and generalized quantile processes," *Stochastic Processes and their Applications*, vol. 69, no. 1, pp. 1–24, Jan. 1997.

[9] H.G. Chew, R.E. Bogner, and C.C. Lim, "Dual-$\nu$ support vector machine with error rate and training size biasing," in *Proc. Int. Conf. Acoustics, Speech, and Signal Proc. (ICASSP)*, Salt Lake City, UT, Jun. 2001.

[10] C. Scott and M. Davenport, "Regression level set estimation via cost-sensitive classification," *IEEE Trans. Signal Processing*, vol. 55, no. 6, pp. 2752–2757, Jun. 2007.