Strategic Classification From Revealed Preferences

Jinshuo Dong Aaron Roth Zachary Schutzman Bo Waggoner Zhiwei Steven Wu

Abstract

We study an online linear classification problem, in which the data is generated by strategic agents who manipulate their features in an effort to change the classification outcome. In rounds, the learner deploys a classifier, and an adversarially chosen agent arrives, possibly manipulating her features to optimally respond to the learner. The learner has no knowledge of the agents' utility functions or "real" features, which may vary widely across agents. Instead, the learner is only able to observe their "revealed preferences" — i.e. the actual manipulated feature vectors they provide. For a broad family of agent cost functions, we give a computationally efficient learning algorithm that is able to obtain diminishing "Stackelberg regret" — a form of policy regret that guarantees that the learner is obtaining loss nearly as small as that of the best classifier in hindsight, even allowing for the fact that agents will best-respond differently to the optimal classifier.

Introduction. Machine learning is typically studied under the assumption that the data distribution a classifier is deployed on is the same as the data distribution it was trained on. However, the outputs of many classification and regression problems are used to make decisions about human beings, such as whether an individual will receive a loan, be hired, be admitted to college, or whether their email will pass through a spam filter. In these settings, the individuals have a vested interest in the outcome, and so the data generating process is better modeled as part of a strategic game in which individuals edit their data to increase the likelihood of a certain outcome. Tax evaders may carefully craft their tax returns to decrease the likelihood of an audit. Home buyers may strategically sign up for more credit cards in an effort to increase their credit score. Email spammers may modify their emails in order to evade existing filters. In each of these settings, the individuals have a natural objective that they want to maximize — they want to increase their probability of being (say) positively classified. However, they also experience a cost from performing these manipulations (tax evaders may have to pay *some* tax to avoid an audit, and email spammers must balance their ability to evade spam filters with their original goal in crafting email text). These costs can be naturally modeled as the distance between the "true" features x of the individual and the manipulated features x' that he ends up sending, according to some measure. In settings of this sort, learning can be viewed as a game between a learner and the set of individuals who generate the data, and the goal of the learner is to compute an equilibrium strategy of the game (according to some notion of equilibrium) that maximizes her utility.

The relevant notion of equilibrium depends on the order of information revelation in the game. Frequently, the learner will first deploy her classifier, and then the data generating players (agents) will get to craft their data with knowledge of the learner's classifier. In a setting like this, the learner should seek to play a *Stackelberg equilibrium* of the game — i.e. she should deploy the classifier that minimizes her error *after* the agents are given an opportunity to best respond to the learner's classifier. This is the approach taken by the most closely related prior work: Brückner and Scheffer (2011) and Hardt et al. (2016). Both of these papers consider a one-shot game and study how to compute the Stackelberg equilibria of this game. To do this, they necessarily assume that the learner has (almost) full knowledge of the agents' utility functions; in particular, it is assumed that the learner has access

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

to the "true" distribution of agent features (before manipulation), and that the costs experienced by the agents for manipulating their data are the same for all agents and known to the learner[1].

This Work: Model. We depart from previous work in two ways. (1) The learner does not know the utility functions of the agents: neither their true features x, nor the cost they experience for manipulation (which can now differ for each agent): instead, all she can observe are their "revealed preferences" – i.e. their strategic responses to her deployed classifier. (2) The agents arrive online, and the learner's goal is to minimize *regret*. We add one further twist. Previous work on strategic classification has typically assumed that all agents are strategic. However, the equilibrium solutions that result from this assumption may be undesirable. For example, in a spam classification setting, the Stackelberg-optimal classifier may attain its optimal accuracy only if all agents — even legitimate (non-spam) senders — actively seek to manipulate their emails to avoid the spam filter. In these settings, it would be more desirable to compute a classifier that was optimal under the assumption that spammers would attempt to manipulate their emails in order to game the classifier, but that did not assume legitimate senders would. To capture this nuance, in our model, only agents whose true label $y_t = -1$ (e.g. spammers) are strategic, and agents for whom $y_t = 1$ are non-strategic.

Formally, in each round $t = 1, \ldots, n$, the learner proposes a classifier β_t. Then, an agent arrives, with some unknown "true" feature vector $x_t \in \mathbb{R}^d$, a label $y_t \in \{\pm 1\}$, and some unknown cost function $d_t : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ for manipulation. If $y_t = 1$, we suppose the agent is nonstrategic and does not manipulate; the algorithm observes (x_t, y_t) and suffers a loss $\ell(\beta_t, x_t, y_t)$. Otherwise, the agent best-responds by selecting a manipulated set of features $\hat{x}_t \in \arg\max_{\hat{x}} \beta_t \cdot \hat{x} - d_t(x_t, \hat{x})$. The algorithm only observes $(\hat{x}_t, y_t)$ and suffers a loss $\ell(\beta_t, \hat{x}_t, y_t)$. In this sense, the algorithm must learn to classify while observing only the "revealed preferences" of the strategic agents. We specifically focus on logistic loss, $\ell(\beta, x, y) = \log\left(1 + e^{-y\beta \cdot x}\right)$, and hinge loss, $\ell(\beta, x, y) = \max\{0, 1 - y\beta \cdot x\}$.

We measure the performance of our algorithms via a quantity that we call *Stackelberg regret*: informally, by comparing the average loss of the learner to the loss she could have experienced with the best fixed classifier in hindsight, *taking into account that the agents would have best-responded differently had she used a different classifier*. If the learner were in fact interacting with the same agent repeatedly, or if the agents (x_t, d_t) were drawn from a fixed distribution, then the guarantee of diminishing Stackelberg regret would imply the convergence to a Stackelberg equilibrium of the corresponding one-shot game. However, Stackelberg regret is more general, and applies even to settings in which the agents are adversarially chosen. Formally, given a sequence of play, the Stackelberg regret of an algorithm is:

$$\mathcal{R}_S = \sum_{t=1}^{n} \ell(\beta_t, \hat{x}_t(\beta_t), y_t) - \min_{\beta} \sum_{t=1}^{n} \ell(\beta, \hat{x}_t(\beta), y_t)$$

This notion of regret has also appeared in the context of online security games Balcan et al. (2015).

This Work: Results. We seek efficient algorithms for minimizing Stackelberg regret. The learner's problem is a bi-level optimization problem in which the objective of the inner layer (the agents' maximization problem) is unknown. Even with full information, bi-level optimization problems are often NP-hard. As a first step in our solution, we seek to identify conditions under which the learner's optimization problem is convex, allowing the learner to apply e.g. online gradient descent. For the second step, we investigate efficient algorithms for minimizing these convex losses, in our limited feedback model.

Convex learning problem. We study learners who deploy *linear classifiers* β_t, and consider two natural learner loss functions: logistic loss (corresponding to logistic regression) and hinge loss (corresponding to support vector machines). Recall that strategic agents are assumed to choose manipulations $\hat{x}_t(\beta_t) = \arg\max_{\hat{x}} \beta_t \cdot \hat{x} - d_t(x_t, \hat{x})$. Using tools from convex analysis, we give general conditions on the cost functions d_t that suffice to make the learner's objective convex, for both logistic and hinge loss, for all x_t. These conditions are satisfied by (among other classes of cost

[1] The particulars of the models studied in Brückner and Scheffer (2011) and Hardt et al. (2016) differ. Brückner and Scheffer model a single data generation player who manipulates the data distribution, and experiences cost equal to the squared ℓ_2 distance of his manipulation. Hardt et al. study a model in which each agent can independently manipulate his own data point, but assume that all agents experience cost as a function of the same separable cost function, known to the learner.

functions) any squared Mahalanobis distance and, more generally, by any norm-induced metric raised to a power greater than one.

Theorem. *For any abstract norm $\|\cdot\|$ on $\mathbb{R}^d$ and any $r > 1$, if $d_t(x_t, \hat{x}) = \frac{1}{r}\|x_t - \hat{x}\|^r$, then both logistic loss and hinge loss $\ell(\beta_t, \hat{x}_t(\beta_t), y_t)$ are convex functions of the hypothesis β_t, given that $\hat{x}_t$ is a best-response to β_t when $y_t = -1$ and $\hat{x}_t = x_t$ when $y_t = 1$.*

The proof relies on a series of tools from convex analysis, in particular relying on *homogeneity* of norms to show that $\beta_t \cdot \hat{x}_t(\beta_t)$ is a convex function of β_t when $\hat{x}_t$ is a best-response.

Optimization with limited feedback. Once we have derived conditions under which the learner's optimization problem is convex, we can in principle achieve quickly diminishing Stackelberg regret with any algorithm for online bandit (i.e. zeroth order) optimization that works for adversarialy chosen loss functions. However, we observe that when some of the agents are non-strategic (e.g. the non-spammers), there is additional structure that we can take advantage of. In particular, on rounds for which the agent is non-strategic and the label is $y_t = 1$, the learner can also derive gradients for her loss function, in contrast to rounds on which the agent is strategic, where the learner only has access to zeroth-order feedback. To take advantage of this, we analyze a variant of the bandit convex optimization algorithm of Flaxman et al. (2005) which can make use of both kinds of feedback. The regret bound we obtain interpolates between the bound of Flaxman et al. (2005), obtained when all agents are strategic, and the regret bound of online gradient descent Zinkevich (2003), obtained when no agents are strategic, as a function of the proportion of the observed agents which were strategic.

Theorem. *For any sequence of agents $A = (a_1, \dots, a_n)$, our mixed-feedback learning algorithm will output a sequence of classifiers $B = (\beta_1, \dots, \beta_n)$ such that the expected Stackelberg regret satisfies*

$$\mathbb{E}\left[\mathcal{R}_S(A, B)\right] \leqslant \frac{\eta}{2}[n\theta \cdot \frac{d^2 M^2}{\delta^2} + n(1-\theta)L^2] + \frac{1}{2\eta}R^2 + 3nL\delta + nLR\delta.$$

We use this theorem to optimally select the learning rate η and a "smoothing" parameter δ. This gives regret bounds depending on θ, the proportion of the n agents who are strategic:

$$\mathbb{E}\left[\mathcal{R}_S(A, B)\right] \leqslant \begin{cases} O(\sqrt{n}), & \theta = 0, \\ O(\sqrt{d}n^{3/4}), & \theta = \Omega(1), \\ O(\sqrt{d}n^{1-\frac{1+\gamma}{4}}) & \theta = O(n^{-\gamma}) \end{cases}$$

The extreme cases correspond to always using "first-order" feedback (nonstrategic agents $\implies$ we can compute gradients at each step and obtain $O(\sqrt{n})$ regret) versus "zeroth-order" feedback (cannot compute many gradients and rely on $O(\sqrt{d}n^{3/4})$-regret algorithm).

We observe that, because our other results show that the learning problem is convex, one can also apply other convex optimization algorithms to the problem and obtain somewhat different regret bounds, in particular, trading off a higher dependence on dimension d for an improved dependence on n.

References

Balcan, M., Blum, A., Haghtalab, N., and Procaccia, A. D. (2015). Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 61–78.

Brückner, M. and Scheffer, T. (2011). Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555. ACM.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. (2016). Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 111–122. ACM.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936.