

Generalized Strategic Classification and the Case of Aligned Incentives

Sagi Levanon

Technion - Israel Institute of Technology
Haifa, Israel
sagilevanon@campus.technion.ac.il

Nir Rosenfeld

Technion - Israel Institute of Technology
Haifa, Israel
nirr@cs.technion.ac.il

ABSTRACT

Strategic classification studies learning in settings where self-interested users can strategically modify their features to obtain favorable predictive outcomes. A key working assumption, however, is that “favorable” always means “positive”; this may be appropriate in some applications (e.g., loan approval), but reduces to a fairly narrow view what user interests can be. In this work we argue for a broader perspective on what accounts for strategic user behavior, and propose and study a flexible model of *generalized strategic classification*. Our generalized model subsumes most current models, but includes other novel settings; among these, we identify and target one intriguing sub-class of problems in which the interests of users and the system are *aligned*. For this cooperative setting, we provide an in-depth analysis, and propose a practical learning approach that is effective and efficient. Returning to our fully generalized model, we show how our results and approach can extend to the most general case. We conclude with a set of experiments that empirically demonstrate the utility of our approach.

KEYWORDS

Strategic Classification, Generalized Strategic Classification, Incentive-Aligned Strategic Classification

1 INTRODUCTION

Machine learning is increasingly being used in domains where human users are the subject of prediction. But when users stand to gain from certain predictive outcomes, they may be prone to act in ways that promote the outcomes they desire. A growing recognition of this idea has led to much recent interest in methods that are able to account for how humans respond to learned models. One such line of research considers the problem of *strategic classification* [5, 15], in which users can modify their features (at some cost) to obtain favorable predictions, and the goal in learning is to be robust to such behavior. Common tasks include loan approval, university admissions, and hiring—all examples in which users have incentive to be classified positively, and in which systems must anticipate user behavior in order to predict well. Strategic classification is appealing as a learning problem in that it is simple yet succinctly captures a natural form of tension that can arise between systems and their users. This has made it the focus of many recent works [11, 13, 18, 20, 21, 27, 29, 31].

But despite the elegant way in which it extends standard binary classification, strategic classification remains narrow in the scope

of strategic behavior it permits. A key working assumption is that while the system is interested in correct predictions, users are interested in *positive* predictions. One artifact of this is that learning is essentially restricted to settings in which one outcome is globally “good” for users (e.g., loan approved), while the other is “bad” (e.g., loan denied); another is that users are grimly framed as always acting to “game” the system. But not all outcomes are “good” or “bad”, and not all strategic behavior is gaming. Here we argue for a broader perspective on what constitutes strategic behavior in classification.

Towards this goal, we propose and study the novel learning framework of *generalized strategic classification* (GSC), which includes standard strategic classification (SC) as a special case, as well as other rich problem sub-classes. Our formulation relies on the simple observation that strategic behavior depends on three key elements: (i) what users *know*, (ii) what users *want*, and (iii) how users *use* their knowledge to promote their goals. For example, in standard strategic classification, users know their true features (whereas the system does not); want positive predictions (rather than correct predictions); and are willing to invest effort or resources to obtain them (by applying costly feature modifications). But other forms of knowledge, aims, and means lead to other types of strategic interactions between a system and its users. The power of our framework lies in providing a simple handle for reasoning about the effects of general forms of interaction on learning.

Within the space of GSC problems, of particular interest to us is a sub-class of problems we refer to as *incentive-aligned strategic classification* (IASC). Aligned incentives naturally arise in settings where predictions are intended to assist users; in particular, they are *for* users, rather than *about* users (as in loans, hiring, etc.). Applications in which incentives align are widespread—recommendation systems, search engines, and online marketplaces are all examples in which prediction is provided by the system as a service to its users, and both parties are interested in accurate predictions.

From a modeling perspective, the key difference between SC and IASC is simply that instead of wanting positive predictions, users now want correct predictions—just as the system does. But from a learning perspective, this mild change transforms the learning problem from one of robustness (to gaming) to one of cooperation, and the key challenge in learning now lies in how to best utilize individual strategic behavior to promote collective improvement. Note this has concrete implications for transparency: to successfully promote its own goals, it is in the best interest of the system to clearly communicate its predictive model to users, so that strategic updates are maximally effective (c.f. SC, where the implications of transparency can be ambiguous [13]).

Appears at the 1st Workshop on Learning with Strategic Agents (LSA 2022). Held as part of the Workshops at the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), N. Bishop, M. Han, L. Tran-Thanh, H. Xu, H. Zhang (chairs), May 9–10, 2022, Online.

How should learning be done in incentive-aligned settings? A natural approach would be to simply take the conventional ERM rule and replace the original features, x , with their strategically-modified counterparts, x' . Indeed, this approach is sensible when applied to the *true error*, i.e., the 0/1-loss. However, our first key result shows that the argument breaks once this naive approach is applied to a *proxy loss*, such as the hinge loss. This has concrete practical implications; as we show, this naïve approach is prone to adverse generalization issues, and is unnecessarily computationally complex. As a solution, and with initial focus on a special instance of IASC, we propose a new *strategic hinge loss*, which builds on an adaptation of the notion of margin to strategic settings. The strategic hinge has a simple and differentiable form, and so can be optimized end-to-end. For generalization, we give Rademacher-based bounds showing that the strategic hinge loss enjoys favorable guarantees.

Building on these results, we return to the most general class of GSC problems, and show that the principles underlying our analysis for IASC hold more broadly. We extend both of our contributions to this general case. First, we give a formula for constructing an appropriate strategic hinge proxy for any instance of GSC, this relying on a careful reinterpretation of the definition of margin. We show that, like for IASC, the strategic proxies for several notable sub-classes are simple and differentiable, and have intuitive interpretations. Second, we extend our generalization bounds to hold for GSC in its broadest form; we also give specialized bounds for several sub-classes, and study their relations. Our bounds have a tight connection to, and closely match, standard bounds for the non-strategic case.

We conclude with an empirical section that includes two sets of experiments. The first set of experiments target generalization, and in this way, complement our theoretical findings. Here we evaluate performance on synthetic data and for several types of strategic environments. The second set of experiments consider an elaborate incentive-aligned environment, designed to mimic a recommendation setting in which users act on the basis of private, personalized past experiences. For this setting, we use data based on a real fashion-related dataset. The strategic hinge in this setting does *not* have a simple form; nonetheless, we propose a tractable approach for its optimization. Our results demonstrate that our approach is efficient and effective.

Taken together, our practical and theoretical results suggest that GSC is an appropriate, well-balanced generalization: it is flexible in the strategic behavior it permits and the learning problems that follow, but at the same time, is precise enough to preserve general statistical properties of classification.

1.1 Related work

The literature on strategic classification is growing rapidly. Various formulations of the problem were studied in earlier works [3, 4, 14], but most recent works adopt the core setup of Hardt et al. [15]. Some studies focus on theoretical aspects; for example, Zhang and Conitzer [29] and Sundaram et al. [27] extend VC theory to account for strategic behavior. Other works focus on practical aspects, such

as Levanon and Rosenfeld [21] who propose a differentiable learning framework for strategic classification. Our work includes a blend of both theory and practice.

There have also been efforts to extend strategic classification beyond its original formulation. For example, Ghalme et al. [13] study a setting where users respond based on individually-estimated classifiers, and Jagadeesan et al. [18] consider users having noisy estimates of model parameters. Sundaram et al. [27] make the connection to adversarial learning, and propose a unified model allowing for more general forms of user response. Our framework subsumes these earlier settings, and generalizes beyond.

Several works in the adversarial literature have suggested margin maximization as a means to achieve robustness [8, 30]. Our work extends beyond this idea and studies the general role margins play under strategic behavior, theoretically and methodologically.

Finally, we note there are many works within the broader area of strategic-aware learning, focusing on dynamics and online learning [7, 9, 31], causal effects [1, 17, 23, 24, 26], strategic regression [2, 16, 28], and strategic ranking [22]. These remain outside the scope of our generalized framework for strategic classification.

2 GENERALIZED STRATEGIC CLASSIFICATION

We begin by reviewing the learning setup for SC, and then proceed to present GSC and IASC.

2.1 Strategic classification

Denote by $x \in \mathcal{X} \subseteq \mathbb{R}^d$ features representing user attributes, and by $y \in \mathcal{Y} = \{-1, 1\}$ their corresponding labels. Let D be some unknown joint distribution over $\mathcal{X} \times \mathcal{Y}$. As in standard classification, at *train time* the system is given access to a sample set of m labeled pairs $S = \{(x_i, y_i)\}_{i=1}^m$ sampled iid from D , on which it aims to learn a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ from some function class H . What differs in strategic classification is that at *test time*, once h has been learned h (and deployed), users can *modify* their features $x \mapsto x'$, in a way we explain shortly. The objective of the system is to correctly classify *modified* inputs, and so its goal in learning is to minimize the expected 0/1 loss, $\text{err}(h) = \mathbb{E}_D[\mathbb{1}\{y \neq h(x')\}]$. Learning is therefore successful if it produces a classifier that is robust to strategic modifications.

The way in which users modify their features derives from the utility they can gain from predictive outcomes. In SC, users are willing to modify features if this will help them be classified positively, denoted $\hat{y} = 1$. But modifications are costly, and the cost of applying $x \mapsto x'$ is given by a *cost function* $c(x, x')$, which is assumed to be known to all. Users are modeled as rational utility-maximizers that update their features via the following *best-response mapping*:

$$\Delta_h^{\text{SC}}(x) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} h(x') - c(x, x') \quad (1)$$

Learning then aims to minimize the expected *strategic error*:

$$\underset{h \in H}{\operatorname{argmin}} \mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{SC}}(x)) \neq y\}] \quad (2)$$

2.2 Generalizing strategic classification

The update in Eq. (1) includes three important ingredients: what users know (i.e., x), what they want (i.e., $h(x') = 1$), and how these

two concepts are integrated into action (i.e., maximizing utility minus cost). But the update is *effective* only when the original features x remain *private user information*.¹ Privacy of information is a crucial aspect of strategic classification, and is essentially what makes it distinct from standard classification. Building on this simple observation, we propose a generalized framework permitting users to make use of richer forms of private information, in more varied ways, and towards broader goals.

Concretely, our GSC framework extends SC in two ways. First, we allow users to hold additional, private *side information*, denoted $z \in \mathcal{Z}$ (and redefine D to be a joint distribution over triples (x, z, y)). Objects z can be of any type, and the main distinction between z -objects and x -objects (which are also private) is that while x are used by the system as input to the classifier (and so are subject to modification by users), z are used only by the user, and only for the purpose of computing Δ (and so remain unmodified).

Second, we allow for broader user objectives by replacing $h(x')$ in Eq. (1) with a more general function of *perceived utility*, denoted $\tilde{u}_h(x', z)$, which encodes the utility users *believe* they obtain from x' under the classifier h , and as a function of z . This gives the *generalized response mapping*:

$$\Delta_h^{\text{GSC}}(x) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \tilde{u}_h(x', z) - c(x, x') \quad (3)$$

To see how u and z work together, observe that Eq. (3) provides a formula for how users *act*, meaning that u reflects how utility is *perceived* by users. This perceived utility may, or may not, align with users' *true* utility (which must be independently defined);² note this means Δ^{GSC} is no longer necessarily a best-response. What controls the degree to which true and perceived utility align is how useful z is as side information when integrated into u to form user beliefs regarding value. Side information can be helpful—but also misleading; a simple example is when $z \in \mathbb{R}$ and acts as additive noise, i.e., $\tilde{u}_h(x', z) = u_h(x') + z$ for true utility u_h , as in Random Utility Theory [6]. But Eq. (3) is more general, and can in principle allow for more elaborate forms of user modeling, such as bounded-rational, Bayesian, or behavioral decision models. We next survey some special instances of Eq. (3) which exemplify this idea.

Notable special cases. Standard SC (Eq. (1)) is obtained from Eq. (3) by setting $z = \emptyset$ (i.e., no side information) and $u_h(x', z) = h(x')$. Notice Eq. (1) can be rewritten as:

$$\Delta_h^{\text{SC}}(x) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \mathbb{1}\{h(x') = 1\} - \frac{1}{2}c(x, x') \quad (4)$$

where now $\tilde{u}_h(x, z) = \mathbb{1}\{h(x') = 1\} = \frac{h(x')+1}{2}$. This simply emphasizes that users want positive predictions, but makes it easy to consider other forms of user interests. For example, for *adversarial* users, we plug in y and obtain:

$$\Delta_h^{\text{adv}}(x, y) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \mathbb{1}\{h(x') \neq y\} - \frac{1}{2}c(x, x') \quad (5)$$

which is made possible once we set $z = y$ as private information. More generally, if we let $z \in \{-1, 1\}$ hold arbitrary values, then we

¹Indeed, had the system known x at test time, any modification would have been futile (and costly).

²In SC, they are the same.

recover the general-preference (GP) model studied in [27]:

$$\Delta_h^{\text{GP}}(x, z) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \mathbb{1}\{h(x') = z\} - \frac{1}{2}c(x, x') \quad (6)$$

here, knowing z is essential for users to effectively respond.

Returning to SC, the recent work of [18] on alternative micro-foundations proposes a noisy response model. This can be recovered by setting $z \in \mathbb{R}^d$ as the random noise term, which gives:

$$\Delta_h^{\text{noise}}(x, z) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} h_{\theta+z}(x') - c(x, x') \quad (7)$$

where $\theta \in \mathbb{R}^d$ are the parameters of the classifier. A more elaborate form of side information can be found in [13], where users are ‘in the dark’ and respond based on an approximate classifier, \tilde{h} , estimated from data:

$$\Delta_h^{\text{dark}}(x, z) \triangleq \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \tilde{h}(x'; z, h) - c(x, x') \quad (8)$$

Here, $\tilde{h}(\cdot; z, h)$ is estimated by users from a sample set $\{(x_j, \hat{y}_j)\}_{j=1}^n$ consisting of additional examples $z = \{x_j\}_{j=1}^n$ labeled by the true classifier, $\hat{y}_j = h(x_j)$. In both of the above, users seek positive predictions, but act based on how they *perceive* utility, as determined by u and z .

2.3 Learning in GSC

In its most general form, the GSC learning objective is:

$$\underset{h \in H}{\operatorname{argmin}} \mathbb{E}_D [\mathbb{1}\{h(\Delta_h^{\text{GSC}}(x, z)) \neq y\}] \quad (9)$$

where for each sub-class of problems, Δ^{GSC} is replaced with its appropriate counterpart. One key difference between Eq. (9) and Eq. (2) is that the former now includes side information z . This requires us to be precise about its role. Clearly, at test time, neither x nor z are observed by the system. However, at train time, it remains a question whether z should be observed or not. For example, in the GP setting of [27], z is assumed to be *known*; in the noisy-parameter setting of [18], when z models users' misperceptions of θ , it is reasonable to consider it as *unknown*; while for the ‘in-the-dark’ setting of [13], it is shown that for known z , learning reduces to standard strategic classification, whereas for unknown z , errors in estimating user responses can blow up in a way that can cause learning to fail completely. Hence, the decision regarding observing z remains task-specific.

In this paper, we focus on the setting where z is known—this places emphasis on the introduction of z into the response mapping Δ , and abstracts away other aspects (such as what system knows or does not). Note also that when z is unknown, from the system's perspective, the user's response mapping Δ (Eq. (3)) is not well-defined, and so requires additional assumptions on how the system compensates for this lack of knowledge in learning, which we aim to avoid.

3 INCENTIVE-ALIGNED STRATEGIC CLASSIFICATION

Next, we introduce and study a particular sub-class of problems in GSC we refer to as *incentive-aligned strategic classification*. Our results here will serve as a basis for our more general results for

GSC in Sec. 4, but are of independent interest (full proofs can be found in the Appendix).

Intuitively, incentive alignment occurs when the utility of users is similar enough to that of the system so that strategic behavior improves both. This can happen when users also seek correct predictions, and the system provides such predictions as a service (recommendation systems are one such example). When incentives align, strategic behavior can improve outcomes for both the system and its users.

We begin with formal definition of incentive alignment (IA).

Definition 1 (Incentive-alignment). Let D be a joint distribution over $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$, H be a function class, and Δ a response mapping. We say the learning task is *incentive-aligned* if $\exists h \in H$ such that $\forall h' \in H$, it holds that:

$$\mathbb{E}_D[\mathbb{1}\{h(\Delta_h(x; z)) \neq y\}] \leq \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}] \quad (10)$$

If such an h exists, we say it *aligns incentives*.

Incentive alignment holds when there is some classifier whose performance on strategically modified inputs is better than the optimal classifier on unmodified inputs. We next describe two novel settings in which users seek to be classified accurately, and so incentive alignment is likely.

Noisy labels (NL). In this simple model, side information z consists of a noisy estimate of a user’s true label, denoted $\tilde{y} \in \{\pm 1\}$. Users act as if \tilde{y} is their true label:

$$\Delta_h^{\text{NL}}(x, \tilde{y}) = \underset{x' \in \mathcal{X}}{\operatorname{argmax}} \mathbb{1}\{h(x') = \tilde{y}\} - \frac{1}{2}c(x, x') \quad (11)$$

although in practice, it may be that $\tilde{y} \neq y$. When \tilde{y} and y are sufficiently correlated, and if the system is able to ‘correct’ erroneous user updates, then incentive alignment should hold. Note Δ^{NL} is a special case of Δ^{GP} (Eq. (6)).

Personalized previous experiences (PPE). Here we mimic a recommendation setting in which the system provides users x with personalized (binary) relevance predictions $y \in \{\pm 1\}$ for various items $a \in \mathbb{R}^\ell$. Instances are tuples (x, z, a, y) (D is extended accordingly), and the system aims to predict relevance for user-item pairs, $\hat{y} = h(x', a)$. For modifying features, users have as side information previously experienced items, in the form of a sample set $z = \{(a_j, y_j)\}_{j=1}^n$, where typically $n \ll m$. Here, $\Delta_h^{\text{PPE}}(x, z)$ is:

$$\underset{x' \in \mathcal{X}}{\operatorname{argmax}} \frac{1}{n} \sum_{j=1}^n \mathbb{1}\{h(x', a_j) = y_j\} - \frac{1}{2}c(x, x') \quad (12)$$

Hence, users modify their features to maximize the accuracy of the proposed h on their past preferences.

In the settings above, system and users cooperate to achieve better accuracy, and are complementary in their strength: the system has large data but on the collective of past users, whereas new users have small or noisy data, but personalized for their own preferences. Learning is successful if the system can harness user efforts to improve accuracy for all. In the following sections we focus on the noisy labels model to establish our main results for IASC, which we will extend in Sec. 4 to GSC. We will return to PPE in Sec. 5.

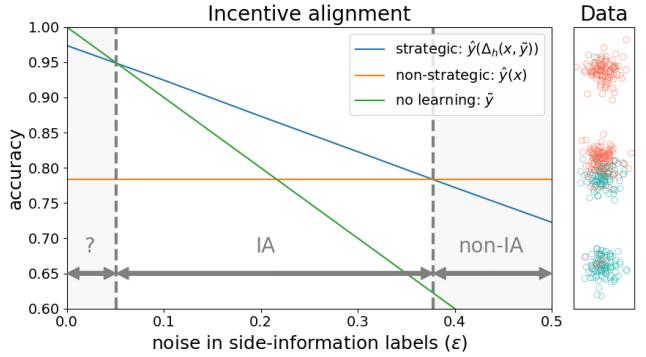


Figure 1: Regions of incentive alignment in NL, for synthetic data (see right plot) and varying noise ϵ . For a wide range of ϵ , learning is incentive-aligned (IA), meaning that the strategic behavior of users is beneficial for both the system and its users. The leftmost region is also IA, but \hat{y} is better than \hat{y} .

3.1 Learning with noisy labels as side information

In this section we study an instance of the noisy-label side-information setting in which noise is uniform and independent: $\tilde{y} = y$ w.p. $1 - \epsilon$ and $\tilde{y} = 1 - y$ w.p. ϵ for all y . The noise parameter ϵ controls how informative \tilde{y} are of y ; this has direct connections to incentive alignment. In Appendix A.1, we characterize the conditions under which incentive alignment holds. Figure 1 shows on synthetic data that incentives align for most values of ϵ (Def. 1), and hence strategic behavior is preferable (details in Appendix B.1).

For learning, we focus on linear classifiers $h_w(x) = \operatorname{sign}(f_w(x))$ where $f_w(x) = w^\top x + b$, and on 2-norm costs $c(x, x') = \|x - x'\|_2$. We use Δ_h to mean Δ_{h_w} , and omit the intercept term b throughout for clarity. A natural approach to minimizing the expected strategic loss (Eq. (9) with Δ^{NL}) is to instead aim at minimizing the empirical risk. For non-strategic data, this approach is well-motivated *in principle*; but in practice, since the 0/1 loss is intractable, a proxy loss is needed, and to control for overfitting, regularization is added. Using the hinge loss as a proxy and L_2 regularization (as in SVM), the objective is:

$$\underset{w}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^m \max\{0, 1 - y_i w^\top x_i\} + \lambda \|w\|_2^2 \quad (13)$$

For strategic settings, adapting Eq. (13) seems appealing and straightforward: simply replace x with $\Delta_h^{\text{NL}}(x, z)$. However, as we show in the next section, this naïve approach is not only difficult to optimize, but may also fail to generalize well. Hence, while minimizing the strategic empirical 0/1 loss is a sound approach, conventional proxy approaches may not be.

The problem with margins. In non-strategic settings, and when data is linearly separable, multiple classifiers can exist having zero empirical error. But some solutions are better than others, and so to guarantee good generalization, additional criteria (beyond loss minimization) must be considered. One popular approach is *max-margin learning*, in which the chosen classifier has the largest

margin: $\gamma_w \triangleq \min_{i \in [m]} y_i \bar{w}^\top x_i$ for $\bar{w} = w/\|w\|$. The main motivation is that large-margin classifiers are known to generalize well, and the canonical max-margin approach is Hard-SVM. A naïve adaptation of Hard-SVM to our strategic setting would be to replace each x_i with $\Delta_h(x_i, \tilde{y}_i)$, this giving:

$$\begin{aligned} \operatorname{argmax}_{w: \|w\|=1} & \min_{i \in [m]} |w^\top \Delta_h(x_i, \tilde{y}_i)| \\ \text{s.t. } & y_i w^\top \Delta_h(x_i, \tilde{y}_i) > 0 \quad \forall i \in [m] \end{aligned} \quad (14)$$

Note that since points now ‘move’, data that was originally separable may no longer satisfy the constraints in Eq. (14). This calls for a notion of *strategic linear separability*.

Definition 2. A sample set is *strategically linearly separable* if there exists a w that separates the data *after* it has been modified, i.e., $y_i w^\top \Delta_h(x_i; z_i) > 0 \forall i \in [m]$.

A key observation is that for the noisy-label side information setting when data is strategically linearly separable, but not linearly separable, the maximal margin is always zero. In these cases, *multiple optimal solutions are likely to exist*, to which the learning algorithm is oblivious. Thus, the naïve approach resurfaces the very same issues that the max-margin approach aims to *avoid*. An illustration of this phenomenon can be found in Appendix 5.

Strategic margins and hinge. Our next result shows that to recover the generalization properties of the max-margin approach, we must rethink the notion of ‘margin’. We begin by reinterpreting the strategic constraints in Eq. (14).

Lemma 1. For any linear classifier w , it holds that:

$$y_i w^\top \Delta_h^{\text{NL}}(x_i, \tilde{y}_i) > 0 \Leftrightarrow y_i (w^\top x_i + 2\tilde{y}_i \|w\|) > 0$$

Proof in Appendix E.5. The constraints can therefore be rewritten in a simple form that does not rely on Δ . This suggests that margins should be measured accordingly.

Definition 3. The *strategic margin* (or *s-margin*) of w is:

$$\gamma_w^{\text{NL}} \triangleq \min_{i \in [m]} y_i (\bar{w}^\top x_i + 2\tilde{y}_i), \quad \bar{w} = w/\|w\| \quad (15)$$

Note that Eq. (15) differs from the standard margin in the corrective term $+2\tilde{y}_i$, which capture the maximal cost and the direction of change. This new margin has an intuitive interpretation: the distance of each x_i is measured w.r.t. an ‘individualized’ classifier, $w^{(i)}$, which is simply w shifted parallelly by 2 units in the direction of \tilde{y}_i (see Fig. 7 in Appendix C). This new margin is *not* equivalent to the naïve objective in Eq. (14).

For strategically-separable data, our approach will be to optimize the s-margin using the appropriate constraints. For non-separable problems, this can easily be extended to the ‘soft’ case using slack variables, akin to standard soft SVM (see Appendix Sec. D.2). From this, we can derive an appropriate proxy loss—the *strategic hinge*, or *s-hinge*.

Definition 4. For noisy labels, the *strategic hinge* is:

$$L_{\text{s-hinge}}^{\text{NL}}(x, \tilde{y}, y; w) \triangleq \max\{0, 1 - y(w^\top x + 2\tilde{y}\|w\|)\}$$

Hence, the s-hinge is simply the standard hinge, but with an additional additive term of $-2y\tilde{y}\|w\|$. To see where this comes

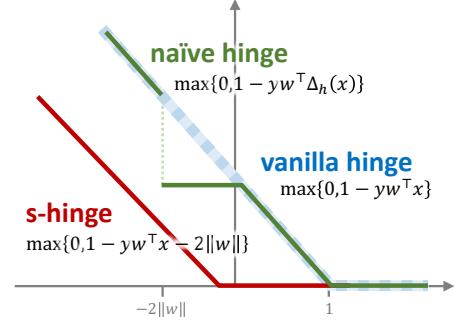


Figure 2: An illustration of the s-hinge for NL with $\tilde{y} = y$. Strategic movement causes the naïve hinge to be flat in $[-2\|w\|, 0]$, leading to undesired indifference. The s-hinge corrects for this.

from, note that the s-hinge can be rewritten as:

$$\max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) - (2 - c(x, \Delta_h^{\text{NL}}(x, \tilde{y})))y\tilde{y}\|w\|\}$$

(proof in Appendix E.8). This is exactly the *naïve hinge*, $\max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y})\}$, but with a term that accounts for both the *actual* and *maximal* cost. Figure 2 illustrates the crux of the naïve hinge, and the s-hinge correction.

Finally, adding regularization gives our learning objective:

$$\operatorname{argmin}_{w \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m L_{\text{s-hinge}}^{\text{NL}}(x_i, \tilde{y}_i, y_i; w) + \lambda \|w\|_2^2 \quad (16)$$

This approach has two key benefits. First, from a computational perspective, since it is differentiable and does not explicitly rely on Δ (which can be discontinuous), learning can be done end-to-end using standard gradient methods. Second, from a statistical perspective, our approach enjoys favorable generalization guarantees. We show this in Sec. 4.2.

4 RESULTS FOR GSC

In this section, we generalize our approach and results to GSC, this using the principles underlying our results for NL in Sec. 3. Our focus remains on linear classifiers, but we will now allow for general side information z , utilities u , and costs c . In particular, our results here cover the personalized previous experiences model (PPE) proposed in Sec. 3.

4.1 Generalized s-hinge

Returning to the standard hinge, note it can be rewritten as $\max\{0, 1 - y \operatorname{sign}(w^\top x)|w^\top x|\}$, which decouples into a ‘correctness’ term, $y \operatorname{sign}(w^\top x) = \mathbb{1}\{y = h(x)\}$, and an un-normalized ‘distance’ term, $d(x; w) \triangleq |w^\top x| = |\bar{w}^\top x| \|w\|$. To adapt the hinge to a strategic setting, one alternative is to simply consider correctness and distance of modified inputs: $\mathbb{1}\{y = h(\Delta_h(x, z))\}$ and $|w^\top \Delta_h(x, z)|$, respectively. This, however, recovers the ‘naïve’ hinge, $\max\{0, 1 - yw^\top \Delta_h(x, z)\}$, which is sub-optimal (Sec. 3.1).

As an alternative, we can rethink the definition of margin, and reinterpret of $d(x; w)$ as *the distance from x to the closest point x' that is classified differently*, i.e., $h(x) \neq h(x')$. In strategic setting,

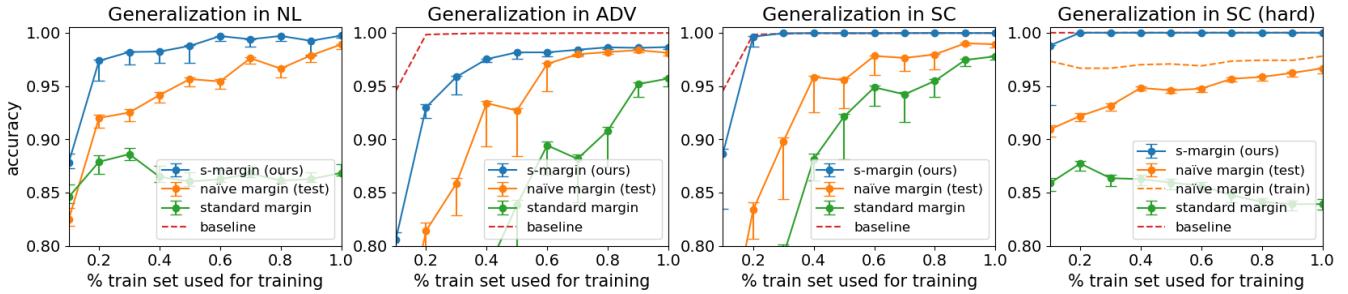


Figure 3: Results for the generalization experiment. The three leftmost plots are on ‘easy’ environments that reflect differences in generalization behavior. The rightmost plot is on a ‘hard’ environment in which the naïve approach also suffers from optimization issues.

the classifier operates on modified inputs, prompting a new notion of *strategic distance*:

$$\begin{aligned} d_\Delta(x, z; w) &= \min_{x'} \|x - x'\| \\ \text{s.t. } h(\Delta_h(x, z)) &\neq h(\Delta_h(x', z)) \end{aligned} \quad (17)$$

(note that, unlike the naïve approach, distance remains to be measured on *un-modified* inputs). The *generalized strategic margin* can be defined using strategic distances, this giving rise to the *generalized strategic hinge*, or *gs-hinge*:

Definition 5. The *generalized strategic hinge* is defined as:

$$\begin{aligned} L_{\text{s-hinge}}^{\text{GSC}}(x, z, y; w) &\triangleq \\ \max\{0, 1 - y \operatorname{sign}(w^\top \Delta_h(x, z)) d_\Delta(x, z; w) \|w\|\} \end{aligned} \quad (18)$$

(for an illustration, see Fig. 8 in Appendix C). For the special case of GP and for norm costs, the strategic distance can be simplified to $d_\Delta^{\text{GP}}(x, z; w) = |\bar{w}^\top x + 2z|$ since points move at most distance 2 in the direction of z (proof in Appendix E.9). For this case, Eq. (18) becomes:

$$L_{\text{s-hinge}}^{\text{GP}}(x, z, y; w) \triangleq \max\{0, 1 - y w^\top x - 2zy \|w\|\}$$

For NL, $z = \tilde{y}$, which recovers $L_{\text{s-hinge}}^{\text{NL}}$, and for SC and adversarial, $z = 1$ and $z = -y$, respectively. For all of the above, our generalized hinge accounts for strategic movement but does not explicitly include $\Delta_h(x, z)$ as a term, which is useful for optimization purposes. Finally, note that for the non-strategic setting, plugging $\Delta_h(x, z) = x$ into Eqs. (17) and (18) recovers the standard margin and hinge.

4.2 Generalization bounds

Our next result provides a data-dependent bound on the expected 0/1 error when minimizing the gs-hinge objective.

Theorem 1. Let D be a joint distribution over triples (x, z, y) , and let $r = \max_{x \in \mathcal{X}} \|x\|$. Denote by \hat{w} the minimizer of the empirical generalized strategic hinge loss with L_2 regularization. Then for every $\delta \in [0, 1]$, if the training set \mathcal{S} includes m samples, then w.p. $\geq 1 - \delta$ it holds that:

$$\mathcal{L}_{0/1} \leq \hat{\mathcal{L}}_{\text{s-hinge}}^{\text{GSC}} + \frac{8r \|\hat{w}\|}{\sqrt{m}} + (1 + 2\rho^{\text{GSC}} \|w\|) \sqrt{\frac{2 \ln(\frac{4\|\hat{w}\|}{\delta})}{m}}$$

where $\mathcal{L}_{0/1}$ is the expected 0/1 loss, $\hat{\mathcal{L}}_{\text{s-hinge}}^{\text{GSC}}$ is the empirical s-hinge loss on \mathcal{S} , and $\rho^{\text{GSC}} = 2r$.

The proof (Appendix E.10) relies on Rademacher bounds, and carefully adapts the approach in [12] to account for strategic updates. The bound in Thm. 1 closely matches the original bound for non-strategic settings, with the only differences being that in the original bound the constant in the middle summand is 4 (vs. 8), and $\rho = r$. We view this as suggesting that, in some sense, the gs-hinge is an appropriate strategic generalization of the standard hinge.

For notable subclasses of GSC, the bound in Thm. 1 can be tightened: the middle constant is reduced to the original 4, and each case has its own ρ term (Appendix E.11). For GP, we get $\rho^{\text{GP}} = (r+2) \leq \rho^{\text{GSC}}$, and also $\rho^{\text{SC}} = \rho^{\text{adv}} = \rho^{\text{GP}}$ in the worst case. For NL, we get $\rho_\epsilon^{\text{NL}} = r - 2 + 4\epsilon$; hence, for $\epsilon = 1/2$, the standard bound is recovered, and for smaller $\epsilon \leq 1/2$, sample complexity in NL is *better* since $\rho_\epsilon^{\text{NL}} < \rho^{\text{GSC}}$ ³ and when incentives align, the empirical loss is also likely to be lower. Together, and for $\bar{\epsilon} \geq 1/2$, we get:

$$\rho_\epsilon^{\text{NL}} \leq \rho \leq \rho_{\bar{\epsilon}}^{\text{NL}} \leq \rho^{\text{SC}} = \rho^{\text{adv}} = \rho^{\text{GP}} \leq \rho^{\text{GSC}}$$

5 EXPERIMENTS

We now turn to our experimental evaluation. Our first experiment empirically studies generalization behavior for the gs-hinge on synthetic data and in various strategic settings. Our second experiment returns to the personalized previous experiences (PPE) setting from Sec. 3 and uses real data.

5.1 Generalization

To complement our theoretical results, we study the generalization behavior of learning with the gs-hinge, as it compares to the naïve hinge and standard (non-strategic) hinge, across multiple settings: NL ($\epsilon = 0$), SC, and adversarial. We also add as a baseline the standard hinge applied to non-strategic data. The challenge in empirically analyzing generalization is that it is difficult to decouple statistical and optimizational issues; this is particularly true in our case, the gs-hinge (for the settings we consider) has both

³Since points move within \mathcal{X} , and since movement is for distance of at most 2, we make the simplifying assumption that $r \geq 2$.

⁴For larger $\epsilon \in (1/2, 1]$, where most labels are flipped, sample complexity is worse. Note $\epsilon = 1$ matches an adversarial setting.

a simpler functional form and better theoretical guarantees than the naïve hinge. We therefore experiment in two types of synthetic environments: ‘easy’ and ‘hard’, which vary in difficulty to optimize. Further details in Appendix B.2.

Results. Figure 3 presents our results. Each plot shows performance for an increasing number of samples used for training. Results are averaged over 30 random splits, with bars showing (asymmetric) standard errors. For the easy environment, the figure shows that across all settings, both our gs-hinge and the naïve hinge achieve roughly optimal accuracy when training on 100% of the data; however the gs-hinge strictly dominates the naïve hinge, and converges significantly faster. This result suggests that the gs-hinge generalizes better. For the hard environment, results show that the naïve approach does not reach the optimum; train accuracy shows that this is not due to overfitting. This shows that the simplicity of the gs-hinge is beneficial.

5.2 Personalized Previous Experiences

In PPE, the goal is to predict for a user-item pair (x, a) whether user x will like item a . Users can modify their inputs based on a (private) sample set of previous experiences, $z = \{(a_j, y_j)\}_{j=1}^n$. We use $c(x, x') = \|x - x'\|_2^2$ and bi-linear classifiers $h_W(x, a) = \text{sign}(a^\top W x)$, where $w_x = Wx$ is a ‘personalized’ classifier.

The PPE setting is unique in that, in essence, users also aim to solve a classification problem. To see this, note that for a given W , user responses $\Delta_h^{\text{PPE}}(x, z)$ (Eq. 12) becomes:

$$\underset{x' \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \sum_{j=1}^n \mathbb{1}\{\text{sign}(a_j^\top W x') \neq y_j\} - \frac{1}{2} \|x - x'\|_2^2$$

This is a linear classification problem: x' act as learned classifier weights, $a_j^{(W)} = a_j^\top W \in \mathbb{R}^d$ serve as features, y_j are target labels, and x' is regularized towards x . In this sense, PPE becomes a problem of multiple agents solving interwoven classification problems, and with similar goals.

Data. We use the Coats Shopping dataset from Schnabel et al. [25]. The data includes 290 users and 300 items (coats), with features for both (e.g., fashion preferences for users, type and color for coats). Labels include relevance scores (here, binarized) given by each user to 40 coats.

Optimization. In principle, both system and users aim to solve difficult 0/1-loss objectives; as these are in general intractable, we model both entities as solving proxy objectives instead. For the system we use the gs-hinge in (Eq. (18)). In PPE, the general hinge is unlikely to have a closed-form solution. This makes learning especially challenging, as it requires solving a triple-nested optimization problem (the outer argmax on w ; the interim min in $d_\Delta(x, z; w)$; and the inner argmax in Δ_h). Nonetheless, we show that if the system uses a squared-loss proxy, the objective becomes tractable (for a full derivation see Appendix Sec. B.3).

Results. We consider three user types that differ by the proxy loss they employ to solve Δ^{PPE} : square, hinge, and logistic. For each user type, we compare the performance of our gs-hinge approach to the standard hinge, and vary the number of previous experiences, $n = |z|$. Figure 4 shows that, for all user types, and with as little as

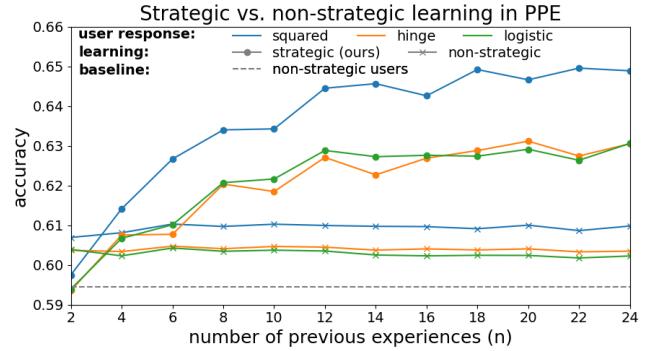


Figure 4: Results for the PPE setting on the coats dataset. Learning with the strategic gs-hinge is significantly better than the standard hinge, starting at $n = 4$ and improving as n increases. Performance is optimal when user respond with a proxy loss that match the system’s (squared). Results show clear incentive alignment.

$n = 4$ examples, the gs-hinge provides a clear improvement over the standard hinge. As n grows, the gain in accuracy becomes more pronounced. For the square-loss user, $n = 24$ improves upon the standard hinge by 6.4%. Improvement exhibits a clear diminishing-returns trend, with most of the gain achieved by $n = 12$ (5.7%). For the hinge- and log-loss users, gains are less pronounced, but still significant (3.8% at $n = 12$, after which gains plateau). Note that both parties are better off when users employ the squared loss. Hence, it is in the best interest of the system to be transparent about the loss it uses, and in the best interest of users follow suit. This is true since incentives align; results show clear incentive-alignment from $n = 4$.

6 CONCLUSIONS

In a world where decisions about humans are increasingly being made by (or with the support of) learned classifiers, it is only reasonable to expect that humans will act to promote their own goals. Our main takeaway is that these goals, and the means taken to pursue them, are varied. Our generalized framework aims to capture this idea, and to provide what we feel is a much needed flexibility in user modeling within this domain. Methodologically, our paper shows that coping with strategic behavior cannot always be done by simply plugging a human response model into exiting methods for non-strategic learning, and that any adaptation of conventional approaches must be done with care.

While our framework is general, it is intentionally restricted to the fundamental setting of supervised binary classification. Our hopes are twofold: that within the limits of our framework, its flexibility will aid in developing new and interesting strategic learning problems; and that beyond our framework’s boundaries, these new problems can extend to broader settings including dynamics, causality, and others.

REFERENCES

- [1] Yahav Bechavod, Katrina Ligett, Steven Wu, and Juba Ziani. 2021. Gaming helps! learning from strategic interactions in natural dynamics. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1234–1242.
- [2] Yahav Bechavod, Chara Podimata, Zhiwei Steven Wu, and Juba Ziani. 2021. Information Discrepancy in Strategic Learning. *arXiv preprint arXiv:2103.01028* (2021).
- [3] Michael Brückner, Christian Kanzow, and Tobias Scheffer. 2012. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research* 13, 1 (2012), 2617–2654.
- [4] Michael Brückner and Tobias Scheffer. 2009. Nash equilibria of static prediction games. In *Advances in neural information processing systems*, 171–179.
- [5] Michael Brückner and Tobias Scheffer. 2011. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 547–555.
- [6] Ennio Cascetta. 2009. *Random Utility Theory*. Springer US, Boston, MA, 89–167. https://doi.org/10.1007/978-0-387-75857-2_3
- [7] Yiling Chen, Yang Liu, and Chara Podimata. 2020. Learning strategy-aware linear classifiers. In *Thirty fourth Conference on Neural Information Processing Systems (NeurIPS), Advances in Neural Information Processing Systems*.
- [8] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. 2020. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In *International Conference on Learning Representations*.
- [9] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. 2018. Strategic Classification from Revealed Preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*. 55–70.
- [10] Murat A. Erdogdu. 2022. Statistical Learning Theory, lecture 6. (2022). <https://erdogdu.github.io/csc2532/lectures/lecture06.pdf>.
- [11] Andrew Estornell, Sanmay Das, Yang Liu, and Yevgeniy Vorobeychik. 2021. Unfairness Despite Awareness: Group-Fair Classification with Strategic Agents. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS), StratML workshop*.
- [12] Ethan Fetaya. 2016. Introduction to Statistical Learning Theory, lecture 7. (2016). https://www.wisdom.weizmann.ac.il/~ethanf/teaching/ItSLT_16/lectures/lec7_no_anim.pdf.
- [13] Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. 2021. Strategic Classification in the Dark. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*.
- [14] Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. 2013. Bayesian games for adversarial regression problems. In *International Conference on Machine Learning*. 55–63.
- [15] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. 2016. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*. 111–122.
- [16] Keegan Harris, Hoda Heidari, and Zhiwei Steven Wu. 2021. Stateful Strategic Regression. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*.
- [17] Keegan Harris, Daniel Ngo, Logan Stapleton, Hoda Heidari, and Zhiwei Steven Wu. 2021. Strategic Instrumental Variable Regression: Recovering Causal Relationships From Strategic Responses. In *Proceedings of the 38th International Conference on Machine Learning (ICML), Workshop on The Neglected Assumptions in Causal Inference*.
- [18] Meena Jagadeesan, Celestine Mendl-Dünner, and Moritz Hardt. 2021. Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*. PMLR, 4687–4697.
- [19] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs.LG]*
- [20] Tosca Lechner and Ruth Urner. 2021. Learning losses for strategic classification. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS), Workshop on Learning in Presence of Strategic Behavior*.
- [21] Sagi Levanon and Nir Rosenfeld. 2021. Strategic Classification Made Practical. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*.
- [22] Lydia T Liu, Nikhil Garg, and Christian Borgs. 2021. Strategic ranking. *arXiv preprint arXiv:2109.08240* (2021).
- [23] John Miller, Smitha Milli, and Moritz Hardt. 2020. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*. PMLR, 6917–6926.
- [24] Nir Rosenfeld, Anna Hilgard, Sai Srivatsa Ravindranath, and David C Parkes. 2020. From Predictions to Decisions: Using Lookahead Regularization. *Advances in Neural Information Processing Systems* 33 (2020).
- [25] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.
- [26] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. 2020. Causal strategic linear regression. In *International Conference on Machine Learning*. PMLR, 8676–8686.
- [27] Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. 2021. PAC-learning for strategic classification. In *International Conference on Machine Learning*. PMLR, 9978–9988.
- [28] Wei Tang, Chien-Ju Ho, and Yang Liu. 2021. Linear models are robust optimal under strategic behavior. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2584–2592.
- [29] Hanrui Zhang and Vincent Conitzer. 2021. Incentive-Aware PAC Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [30] Yan Zhou, Murat Kantarcioğlu, Bhavani Thuraisingham, and Bowei Xi. 2012. Adversarial Support Vector Machine Learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [31] Tijana Zrnic, Eric Mazumdar, Shankar Sastry, and Michael Jordan. 2021. Who Leads and Who Follows in Strategic Classification? *Advances in Neural Information Processing Systems* 34 (2021).

A ADDITIONAL RESULTS

A.1 Conditions for Incentive Alignment in NL

Our first results give a characterization of when a learning task is incentive aligned (see Definition 1) in a simple instance of the noisy label model with independent uniform noise: $\tilde{y} = y$ w.p. $1 - \epsilon$ and $\tilde{y} = 1 - y$ w.p. ϵ for all y .

As the noise parameter ϵ controls how informative \tilde{y} are of y , we can ask: for what values of ϵ is strategic behavior beneficial to all? As we show, the answer directly depends on whether incentives can be aligned.

We begin with a definition of *flipping cost*.

Definition 6. The *flipping cost* of a point x with respect to a classifier h and a cost function c is:

$$\phi_h(x) \triangleq \min_{x' \in \mathcal{X}} c(x, x') \quad \text{s.t. } h(x') \neq h(x) \quad (19)$$

Hence, $\phi_h(x)$ is the minimal cost required for modifying x to ‘flip’ its label (we will return to label flipping in Sec. 4). Next, under the common case of $c(x, x) = 0$, observe that:

Lemma 2.

- If $h(x) = \tilde{y}$, then $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = h(x)$
- If $h(x) \neq \tilde{y}$, then $h(\Delta_h^{\text{NL}}(x, \tilde{y})) \neq h(x)$ iff $\phi_h(x) \leq 2$

Proof in Appendix E.1. Given this, we will define:

$$\Phi_h = \{x : \phi_h(x) \leq 2\} \quad (20)$$

to include all points that *can* flip their label.

We can now state our first result.

Theorem 2. Let D be a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Assume $z = \tilde{y} = y$ with probability $1 - \epsilon$, and $-y$ otherwise, and assume features are modified via Δ^{NL} in Eq. (11). Then it is better for the system to encourage strategic behavior iff it learns a classifier h for which it holds that:

$$\begin{aligned} \mathbb{P}[x \notin \Phi_h \wedge h(x) \neq y] + \mathbb{P}[x \in \Phi_h] \cdot \epsilon \\ \leq \min_{h' \in H} P(h'(x) \neq y) \end{aligned} \quad (21)$$

The proof (Appendix E.2) relies on the coupling of a technical lemma with the careful accounting of different error types. The theorem states that strategic behavior is helpful if the system can learn a classifier whose error on points that cannot flip their label (first summand) plus an ϵ -proportion of the points that can flip their label (second summand) is at most the optimal error with no strategic behavior (RHS). This relates to incentive alignment:

Corollary 1. If such an h exists, then the learning task is incentive-aligned.

Proof in Appendix E.3. Thm. 2 states when it is preferable for users to modify features via Δ^{NL} , rather than simply reporting their original x . For completeness, we also state when predictions \hat{y} on modified inputs are better than \tilde{y} .

Lemma 3. In the same setting as above, it is better for users to use system-provided predictions \hat{y} than their own side-information \tilde{y} iff the system uses a classifier h for which:

$$\frac{\mathbb{P}[x \notin \Phi_h \wedge h(x) \neq y]}{\mathbb{P}[x \notin \Phi_h]} \leq \epsilon \quad (22)$$

Proof in Appendix E.4. Lemma 3 states that predictions are helpful as long as errors on points who cannot flip their label (and so cannot in principle “correct” system errors) is no more than the noise inherent in side information, ϵ . Proofs for both results rely on the coupling of a technical lemma with the careful accounting of different error types. Figure 1 shows on synthetic data that for most values of ϵ , strategic behavior is preferable (details in Appendix B.1).

A.2 NL as an Extreme Subclass

Recall that our discussion regarding the need for a specialize hinge loss for strategic settings began with the observation that, under a naïve approach, multiple classifiers can obtain a margin of zero. Our next result shows that, in this sense, NL is a ‘extreme’ class within problems in GP.

Lemma 4. Let $S = \{(x_i, y_i)\}_{i=1}^m$. Then of all possible assignments to corresponding side information $\{z_i\}_{i=1}^m$, the assignment $z_i = y_i$ for all $i \in [m]$ has the largest set of feasible solutions having zero margin (when such exist).⁵

⁵Note that the set of feasible solutions may also be large; the ratio of zero-margin feasible solutions is data-dependent.

PROOF. Let $S = \{(x_i, z_i, y_i)\}_{i=1}^m$ such that $\exists i \in [m], z_i \neq y_i$.

Denote by $H'_S \subseteq H$ the set of all feasible solutions to the naive S-SVM optimization problem (Eq. 14) for the GP setting.

Denote by $H''_S \subseteq H'$ the set of all feasible solutions which induce zero margin ($\min_{i \in [m]} |w^\top \Delta_h(x_i, z_i)| = 0$).

Fix some $i \in [m]$ for which $z_i \neq y_i$. Consider the sample set $S' = S \setminus \{(x_i, z_i, y_i)\} \cup \{(x_i, y_i, y_i)\}$.

The i_{th} constraint for S' is strictly weaker in comparison to S (Lemma 1). The rest of the constraints remain unchanged.

Therefore $\forall h \in H'_S, h \in H'_{S'}$.

If $h \in H''_S$ than $\min_{i \in [m]} |w^\top \Delta_h(x_i, z_i)| = 0$. This means that $\exists j \in [m], w^\top \Delta_h(x_j, z_j) = 0$. Notice that $j \neq i$. proof:

If $h \in H''_{S'}$ than all constraints must hold. In particular $y_i w^\top x_i < -2$ (Lemma 1). If $w^\top \Delta_h(x_i, z_i) = 0$ than $h(\Delta_h(x_i, z_i)) \neq h(x_i)$ which means that x_i modified it's features to be classified as z_i . Since $z_i \neq y_i$, the i_{th} constraint does not hold - a contradiction. \square

$w^\top \Delta_h(x_j, z_j) = 0 \Rightarrow \min_{i \in [m]} |w^\top \Delta_h(x_i, z_i)| = 0 \Rightarrow h \in H''_{S'}$. We get that $\forall h \in H''_S, h \in H''_{S'}$. Therefore $H''_S \subseteq H''_{S'}$.

Finally, applying such modifications to S recursively we get that H''_S is maximal for $S^* = \{(x_i, y_i, y_i)\}_{i=1}^m$. \square

B ADDITIONAL EXPERIMENTAL DETAILS

Technical details. All experiment results were averaged over 30 randomized experiments (data generation/shuffles and train/test splits). In our experiments we used the soft S-SVM for the naive and non-strategic models and used our proposed soft formulation for the strategically aware models. These methods require a hyper-parameter (λ) controlling the tradeoff between maximization and amount of constraint violations. We tuned this hyper-parameter using cross-validation of 3 splits for $\lambda \in \{0.01, 0.1, 1\}$. For the optimization we used the standard Adam optimizer [19] and a learning rate of 0.05. Each model trained for 200/200/50 epochs with a batch size of 5/16/24 for the generalization experiment, the varying preference noise experiment and the PPE experiment respectively.

B.1 Experiment: Varying ϵ in NL (Sec. 3.1)

In this experiment our goal was to emphasize that there are **simple and common** environments which are incentive-aligned, for both low and high values of ϵ . For this we generated a 2D dataset with clusters sampled from a normal distribution as depicted in Figure 1. Then generated user preferences according to the Noisy labels setting for various ϵ values between 0 and 0.5. For the learning process we used our proposed approach (Eq. 16) and compared its accuracy to the baseline accuracy (non-strategic model) and the pure user-information accuracy ($1 - \epsilon$). Results can be viewed in Figure 1.

Data. For this experiment we considered 2D user features, and the distribution we used is composed out of 4 sub-distributions:

- $(x, y) \sim N\left(\begin{pmatrix} -5 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 20 \end{pmatrix}\right) \times \{1 \text{ w.p. } 0.95, -1 \text{ w.p. } 0.05\}$
- $(x, y) \sim N\left(\begin{pmatrix} 0.3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 20 \end{pmatrix}\right) \times \{1 \text{ w.p. } 0, -1 \text{ w.p. } 1\}$
- $(x, y) \sim N\left(\begin{pmatrix} -0.3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 20 \end{pmatrix}\right) \times \{1 \text{ w.p. } 1, -1 \text{ w.p. } 0\}$
- $(x, y) \sim N\left(\begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 20 \end{pmatrix}\right) \times \{1 \text{ w.p. } 0.05, -1 \text{ w.p. } 0.95\}$

Recall $\tilde{y} \sim \{y \text{ w.p. } 1 - \epsilon, -y \text{ w.p. } \epsilon\}$.

From each distribution we sampled 50 samples for the train set and 1250 samples for the test set.

B.2 Experiment: Generalization (Sec. 5.1)

Optimization. Here we describe how we calculated Δ^{GP} (for the naive approach) in a differential way - allowing GD-based optimization methods. Notice that for norm-based cost function, the GP response mapping argmax problem has a closed-form solution:

$$\Delta_h^{GP}(x, z) = \begin{cases} x - \frac{z w^\top x w}{\|w\|^2} & -2 \leq \frac{z w^\top x}{\|w\|} \leq 0 \\ x & \text{otherwise} \end{cases}$$

This closed form solution is unfortunately non-differentiable. For this reason we apply the following differentiable approximation to Δ^{GP} :

$$\begin{aligned} \Delta_h^{GP}(x, z) &= x - \frac{z w^\top x w}{\|w\|^2} \cdot \sigma(\text{cond}) \\ \text{cond} &= \left(-\frac{z w^\top x}{\|w\|}\right) \cdot \left(\frac{z w^\top x}{\|w\|} + 2\right) \end{aligned}$$

Where σ is the sigmoid function.

Data. For these experiment we considered 2D user features, and the distributions we used are composed out of 2 sub-distributions:

NL: (i) $(x, z, y) \sim N\left(\begin{pmatrix} 10 \\ 0 \end{pmatrix}, \begin{pmatrix} 5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. } 1\} \times \{1 \text{ w.p. } 1\}$, (ii) $(x, z, y) \sim N\left(\begin{pmatrix} -10 \\ 0 \end{pmatrix}, \begin{pmatrix} 5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{-1 \text{ w.p. } 1\} \times \{-1 \text{ w.p. } 1\}$

ADV: (i) $(x, z, y) \sim N\left(\begin{pmatrix} 15.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{-1 \text{ w.p. 1}\} \times \{1 \text{ w.p. 1}\}$, (ii) $(x, z, y) \sim N\left(\begin{pmatrix} 4.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. 1}\} \times \{-1 \text{ w.p. 1}\}$

SC: (i) $(x, z, y) \sim N\left(\begin{pmatrix} 15 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. 1}\} \times \{1 \text{ w.p. 1}\}$, (ii) $(x, z, y) \sim N\left(\begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. 1}\} \times \{-1 \text{ w.p. 1}\}$

SC (hard): (i) $(x, z, y) \sim N\left(\begin{pmatrix} 2.25 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. 1}\} \times \{1 \text{ w.p. 1}\}$, (ii) $(x, z, y) \sim N\left(\begin{pmatrix} -2.25 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5, 0 \\ 0, 0.2 \end{pmatrix}\right) \times \{1 \text{ w.p. 1}\} \times \{-1 \text{ w.p. 1}\}$

From each distribution we sampled 25 samples for the train set and 1250 samples for the test set.

B.3 Experiment: Private Personalized Experiences (PPE) (Sec. 5.2)

Notice that in this setting, Δ does not have a closed-form solution and can only be approximated by **both** users and the system. This means that even at test time, users will modify their features according to an approximation of Δ^{PPE} . To test the effectiveness and robustness of our proposed algorithm, we fix the approximation of the system and vary the approximation of users. We trained the system with the MSE loss, and tested 3 different user approximation losses: MSE loss, classic hinge loss and logistic loss. For each experiment, we varied the amount of items users were exposed to in advanced (size of z). The more items in z , the more accurate their "model" will be, and therefore we expect higher accuracy.

$$\begin{aligned} \tilde{\Delta}_h^{\text{PPE}}(x, z) &= \operatorname{argmin}_{x' \in \mathcal{X}} \frac{1}{|z|} \sum_{(a,y) \in z} L(x', a, y; W) + \frac{1}{2} \|x' - x\|^2 \\ L^{\text{squared}}(x, a, y; W) &= \|a^\top W x - y\|^2 \\ L^{\text{hinge}}(x, a, y; W) &= \max\{0, 1 - y \cdot (a^\top W x)\} \\ L^{\text{logistic}}(x, a, y; W) &= e^{-y \cdot (a^\top W x)} \end{aligned} \quad (23)$$

System training procedure. In PPE the recommender takes 2 arguments (Eq. 12) so we used an adaptation of the GSC s-hinge loss (Eq.18) for this experiment.

$$\begin{aligned} L_{\text{s-hinge}}^{\text{PPE}}(x, z, a, y; W) &\triangleq \\ \max\{0, 1 - y \operatorname{sign}(a^\top W \Delta_h(x, z)) d_\Delta(x, z; W) \|a^\top W\|\} \end{aligned} \quad (24)$$

We begin by replacing the *sign* function with a sigmoid function - an approximated smooth and differential version of *sign*. We continue by describing how we calculate $\tilde{\Delta}_h^{\text{PPE}}(x, z)$. Recall the system uses the *MSE* approximation and therefore $\tilde{\Delta}_h^{\text{PPE}}(x, z)$ can be rewritten as:

$$\begin{aligned} \tilde{\Delta}_h^{\text{PPE}}(x, z) &= \operatorname{argmin}_{x' \in \mathcal{X}} \|AWx' - Y\|^2 + \frac{|z|}{2} \|x' - x\|^2 \\ A &= (a_1, \dots, a_{|z|}), Y = (y_1, \dots, y_{|z|}) \end{aligned} \quad (25)$$

This is a standard LS with regularization optimization problem and has a closed form solution:

$$\tilde{\Delta}_h^{\text{PPE}}(x, z) = (2(AW)^\top AW + |z|\mathbb{I})^{-1}(2(AW)^\top Y + |z|x) \quad (26)$$

Next, we describe how we find $d_\Delta(x, z; W)$. Notice that according to the MSE approximation, users' strategic modifications are linear transformations:

$$\begin{aligned} \tilde{\Delta}_h^{\text{PPE}}(x, z) &= \alpha_{W,z}x + \beta_{W,z} \\ \alpha_{W,z} &= |z|(2(AW)^\top AW + |z|\mathbb{I})^{-1} \\ \beta_{W,z} &= (2(AW)^\top AW + |z|\mathbb{I})^{-1}(2(AW)^\top Y) \end{aligned} \quad (27)$$

Being a linear transformation, it is also continuous. This implies that for any x , the closest x' to it for which $h(\Delta_h(x, z)) \neq h(\Delta_h(x', z))$ (Eq. 17), has to hold $\frac{|aW\Delta_h(x', z)|}{\|aW\|} = 0$. Meaning it's distance to the hyperplane after applying modifications is 0.

Now, we plug in Eq. 27 to this constraint and get:

$$aW\alpha_{W,z}x - aW\beta_{W,z} = 0 \quad (28)$$

All that remains is to solve a convex optimization problem with linear equality constraints to find $d_\Delta(x, z; W)$. To do this in a differential way, we apply another approximation and solve the following optimization problem instead:

$$\tilde{\Delta}_h^{\text{PPE}}(x, z) = \operatorname{argmin}_{x' \in \mathcal{X}} \|aW\alpha_{W,z}x - aW\beta_{W,z}\|^2 + \lambda \|x' - x\|^2 \quad (29)$$

For fairly small values of λ (In our experiments we used $\lambda = 0.01$). This optimization problem too has a closed form solution which concludes the training procedure calculations.

C ADDITIONAL ILLUSTRATIONS

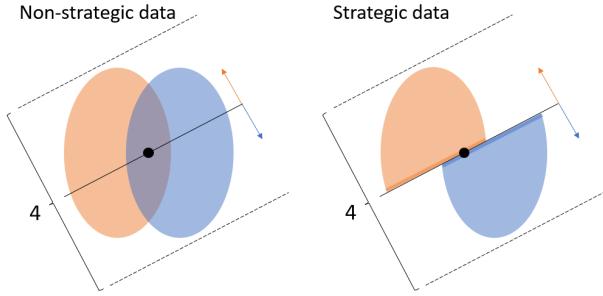


Figure 5: For some environments, strategic behavior induces multiple solutions. An extreme case is the *NL* setting with $\epsilon = 0$. For this setting and the depicted distribution, any separating hyperplane which goes through the center of the distribution (black dot), separates the strategic data perfectly. The reason is that for these classifiers, all users are willing to move to the hyperplane in case they are not classified correctly, since the modification cost is small enough. Also any loss-minimizing solution has a margin of 0 since the distribution is not linearly separable.

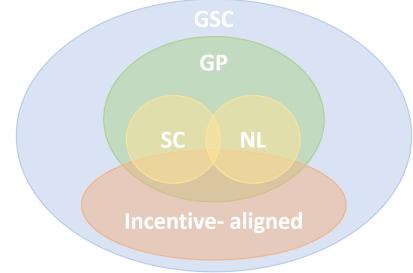


Figure 6: A depiction of the generalized strategic classification world and various sub-classes within it.

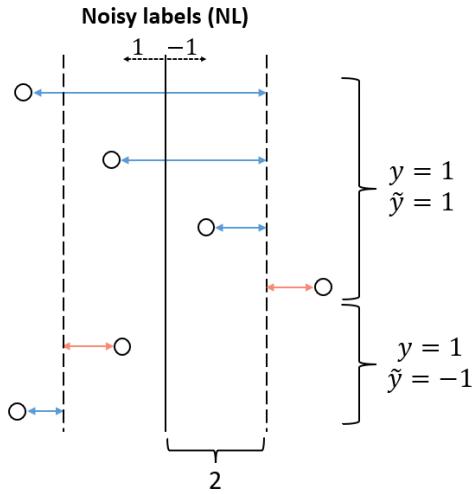


Figure 7: An illustration of the GP/NL s -margin minimization objective for different samples. The hyperplane classifies points to its left as -1 (orange side) and points to its right as $+1$ (blue side). The red and blue lines represent d_Δ - which in this setting is the distances of users to the shifted hyperplane 2 units into their "undesired side". Red lines mean that the user is in a "red" region ($\text{sign}(yw^\top \Delta_h(x; z)) < 0$) and similarly for blue lines ($\text{sign}(yw^\top \Delta_h(x; z)) > 0$).

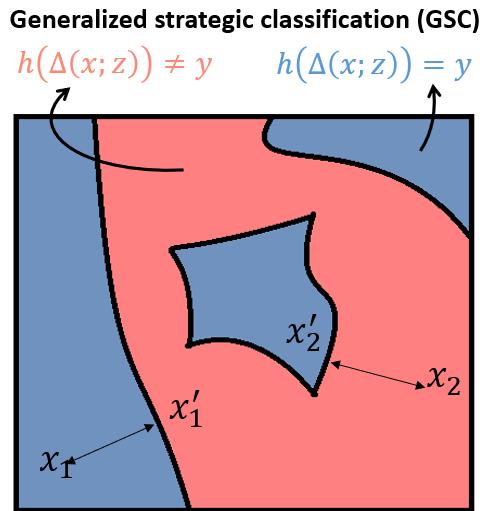


Figure 8: Let h, Δ, z, y be a classifier, a response mapping, user latent information and a label. Notice that these elements induce regions in X for which $h(\Delta(x; z)) = y$ (blue) and $h(\Delta(x; z)) \neq y$ (red). The objective of the gs-hinge (Eq. 18) is to maximize $d_\Delta(x, z; h)$ (Eq. 17) for x values in blue regions, and minimize it for x values in red regions. The reason is that $\text{sign}(yw^\top \Delta_h(x_1; z))d_{\Delta_h}(x_1, z; h) > 0$ and $\text{sign}(yw^\top \Delta_h(x_2; z))d_{\Delta_h}(x_2, z; h) < 0$

D ADDITIONAL DEFINITIONS

D.1 NL Hard formulation equivalent optimization problem

According to Lemma 1, the Hard formulation from Eq. 14 for the NL setting can be rewritten as:

$$\begin{aligned} \operatorname{argmax}_{w: \|w\|=1} \min_{i \in [m]} & |y_i(w^\top x + 2\tilde{y}_i)| \\ \text{s.t. } & y_i(w^\top x + 2\tilde{y}_i) > 0 \quad \forall i \in [m] \end{aligned} \quad (30)$$

This optimization problem can be reformulated as follows:

Algorithm 1 NL Hard formulation

Input: $\{(x_i, \tilde{y}_i, y)\}^m$

Solve:

$$\begin{aligned} w_0 = \operatorname{argmin}_w & \|w\|^2 \\ \text{s.t. } & \forall i \in [m], y_i(w_0^\top x + 2\tilde{y}_i \|w_0\|) \geq 1 \end{aligned}$$

Output: $\hat{w} = \frac{w_0}{\|w_0\|}$

Algorithm 1 and Eq. 30 output the same classifier w . Proof in Appendix E.6.

D.2 NL/GP Soft formulation

The hard constraints of the algorithm 1 optimization problem can be soften by introducing slack variables, similarly to the classic soft SVM algorithm.

Algorithm 2 NL/GP Soft formulation

Input: $\{(x_i, \tilde{y}_i, y)\}^m$

Parameter: $\lambda > 0$

Solve:

$$\begin{aligned} w, \xi_i = \operatorname{argmin}_{w, \xi_i} & \lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \xi_i \\ \text{s.t. } & \forall i \in [m], y_i(w^\top x + 2\tilde{y}_i \|w\|) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \end{aligned}$$

Output: w

We can rewrite the optimization problem of Algorithm 2 as a regularized loss-minimization problem:

$$\begin{aligned} w, \xi_i = \operatorname{argmin}_{w, \xi_i} & \lambda \|w\|^2 + \frac{1}{m} L_{\text{s-hinge}}^{\text{NL}}(x_i, \tilde{y}_i, y_i; w) \\ L_{\text{s-hinge}}^{\text{NL}}(x, \tilde{y}, y; w) \triangleq & \max\{0, 1 - y(w^\top x + 2\tilde{y} \|w\|)\} \end{aligned}$$

Proof in Appendix E.7.

E PROOFS

E.1 Lemma 2

PROOF. Assume $c(x, x) = 0$.

- If $h(x) = \tilde{y}$: Assume, for sake of contradiction, that $h(\Delta_h^{\text{NL}}(x, \tilde{y})) \neq h(x)$.
So $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = -\tilde{y}$.
Recall $\Delta_h^{\text{NL}}(x, \tilde{y}) = \operatorname{argmax}_{x' \in \mathcal{X}} \mathbb{1}\{h(x') = \tilde{y}\} - \frac{1}{2}c(x, x')$.
This means that $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) \leq \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
But
 $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) = 1$.
 $\mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) = \mathbb{1}\{-\tilde{y} = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) = -\frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) \leq 0 < 1$.
So $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) > \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
Therefore the assumption must be false.
- If $h(x) \neq \tilde{y}$: $h(\Delta_h^{\text{NL}}(x, \tilde{y})) \neq h(x)$ iff $\phi_h(x) \leq 2$

- If $h(\Delta_h^{\text{NL}}(x, \tilde{y})) \neq h(x)$:
 - So $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}$.
 - Assume, for sake of contradiction, that $\phi_h(x) > 2$.
 - Therefore $c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) > 2$.
 - Again, $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) \leq \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
 - But $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) = 0$.
 - $\mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) = \mathbb{1}\{\tilde{y} = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) < 1 - \frac{1}{2} \cdot 2 = 0$.
 - So $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) > \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
 - Therefore the assumption must be false.
- If $\phi_h(x) \leq 2$:
 - Let x' be the minimizer of the flipping cost optimization problem (Definition 6).
 - Therefore $h(x) \neq h(x')$ and $c(x, x') \leq 2$. This means that $h(x') = \tilde{y}$.
 - Assume, for sake of contradiction, that $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = h(x)$.
 - So $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = -\tilde{y}$.
 - Again, $\mathbb{1}\{h(x') = \tilde{y}\} - \frac{1}{2}c(x, x') \leq \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
 - But $\mathbb{1}\{h(x') = \tilde{y}\} - \frac{1}{2}c(x, x') = \mathbb{1}\{\tilde{y} = \tilde{y}\} - \frac{1}{2}c(x, x') \geq 1 - \frac{1}{2} \cdot 2 = 0 > -1$.
 - $\mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) = \mathbb{1}\{-\tilde{y} = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) = -1 - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y})) \leq 0$.
 - So $\mathbb{1}\{h(x) = \tilde{y}\} - \frac{1}{2}c(x, x) > \mathbb{1}\{h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y}\} - \frac{1}{2}c(x, \Delta_h^{\text{NL}}(x, \tilde{y}))$.
 - Therefore the assumption must be false.

□

E.2 Theorem 2

PROOF. Let D be a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Assume $z = \tilde{y} = y$ with probability $1 - \epsilon$, and $-y$ otherwise, and assume features are modified via Δ^{NL} in Eq. (11).

It is better for the system to encourage strategic behavior iff it learns a classifier h for which it holds that:

$$\mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] \leq \min_{h' \in H} \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}]$$

- RHS:

$$\mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}] = \mathbb{P}[h'(x) \neq y]$$

- LHS:

From Lemma 2 we derive 2 conclusions:

- 1. $x \in \Phi_h \Rightarrow h(\Delta_h^{\text{NL}}(x; \tilde{y})) = \tilde{y}$.
- 2. $x \notin \Phi_h \Rightarrow h(\Delta_h^{\text{NL}}(x; \tilde{y})) = h(x)$.

$$\begin{aligned} \mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] &= \mathbb{P}[h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y] = \\ \mathbb{P}[h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y | x \in \Phi_h] \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y | x \notin \Phi_h] \cdot \mathbb{P}[x \notin \Phi_h] &=_{1,2} \\ \mathbb{P}[\tilde{y} \neq y | x \in \Phi_h] \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[h(x) \neq y | x \notin \Phi_h] \cdot \mathbb{P}[x \notin \Phi_h] &= \tilde{y} \text{ and } x \text{ are i.i.d} \\ \epsilon \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[h(x) \neq y \wedge x \notin \Phi_h] & \end{aligned}$$

Finally, we get:

$$\begin{aligned} \mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] &\leq \min_{h' \in H} \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}] \\ \Leftrightarrow \\ \epsilon \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[h(x) \neq y \wedge x \notin \Phi_h] &\leq \min_{h' \in H} \mathbb{P}[h'(x) \neq y] \end{aligned}$$

□

E.3 Corollary 1

PROOF. if such an h exists, then according to Theorem 2:

$$\mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] \leq \min_{h' \in H} \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}]$$

Therefore $\exists h \in H$ such that $\forall h' \in H$ it holds that:

$$\mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] \leq \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}]$$

According to Definition 1, the environment is incentive-aligned. \square

E.4 Lemma 3

PROOF. Let D be a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Assume $z = \tilde{y} = y$ with probability $1 - \epsilon$, and $-y$ otherwise, and assume features are modified via Δ_h^{NL} in Eq. (11).

it is better for users to use system-provided predictions \hat{y} than their own side-information \tilde{y} iff the system learns a classifier h for which it holds that:

$$\mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] \leq \mathbb{E}_D[\mathbb{1}\{\tilde{y} \neq y\}]$$

- RHS:

$$\mathbb{E}_D[\mathbb{1}\{\tilde{y} \neq y\}] = \mathbb{P}[\tilde{y} \neq y] = \epsilon$$

- LHS:

$$\begin{aligned} \mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] &= \mathbb{P}[h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y] =_{\text{Theorem 2 proof}} \\ &\epsilon \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[h(x) \neq y \wedge x \notin \Phi_h] \end{aligned}$$

Finally, we get:

$$\begin{aligned} \mathbb{E}_D[\mathbb{1}\{h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq y\}] &\leq \min_{h' \in H} \mathbb{E}_D[\mathbb{1}\{h'(x) \neq y\}] \\ &\Leftrightarrow \\ &\epsilon \cdot \mathbb{P}[x \in \Phi_h] + \mathbb{P}[\tilde{y} \neq y \wedge x \notin \Phi_h] \leq \epsilon \\ &\Leftrightarrow \\ &\mathbb{P}[\tilde{y} \neq y \wedge x \notin \Phi_h] \leq \epsilon(1 - \mathbb{P}[x \in \Phi_h]) \\ &\Leftrightarrow \\ &\mathbb{P}[\tilde{y} \neq y \wedge x \notin \Phi_h] \leq \epsilon \cdot \mathbb{P}[x \notin \Phi_h] \\ &\Leftrightarrow \\ &\frac{\mathbb{P}[\tilde{y} \neq y \wedge x \notin \Phi_h]}{\mathbb{P}[x \notin \Phi_h]} \leq \epsilon \end{aligned}$$

\square

E.5 Lemma 1

PROOF. Let $h = (w)$ be a linear classifier and $(x, \tilde{y}, y) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$. We show that:

$$yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) > 0 \Leftrightarrow y(w^\top x + 2\tilde{y}\|w\|) > 0$$

Or equivalently:

$$h(\Delta_h^{\text{NL}}(x, \tilde{y})) = y \Leftrightarrow \frac{yw^\top x}{\|w\|} > -2\tilde{y}y$$

Observation:

In NL, the cost function is the euclidean distance between two vectors ($c(x, x') = \|x - x'\|$).

Therefore the flipping cost (Definition 6) of a user x is the euclidean distance to the separating hyperplane ($\phi_h(x) = \frac{|w^\top x|}{\|w\|}$).

- If $y = \tilde{y}$:

- If $h(x) = \tilde{y}$:
Than $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y} = y$ (Lemma 2).
And $yw^\top x = \tilde{y}w^\top x > 0 \Leftrightarrow \frac{yw^\top x}{\|w\|} > 0 > -2 = -2\tilde{y}y$
- If $h(x) \neq \tilde{y}$:
Than $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y} = y \Leftrightarrow \phi_h(x) \leq 2$ (Lemma 2).
 $\phi_h(x) \leq 2 \Leftrightarrow \frac{|w^\top x|}{\|w\|} \leq 2$ (Observation).
 $\frac{|w^\top x|}{\|w\|} \leq 2 \Leftrightarrow \frac{\tilde{y}w^\top x}{\|w\|} \geq -2$ ($\tilde{y}w^\top x < 0$).
 $\frac{\tilde{y}w^\top x}{\|w\|} \geq -2 \Leftrightarrow \frac{yw^\top x}{\|w\|} \geq -2\tilde{y}y$

- If $y \neq \tilde{y}$

- If $h(x) = \tilde{y}$:
Than $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y} = -y$ (Lemma 2).
And $yw^\top x = -\tilde{y}w^\top x < 0 \Leftrightarrow \frac{yw^\top x}{\|w\|} < 0 < 2 = -2\tilde{y}y$
- If $h(x) \neq \tilde{y}$:
Than $h(\Delta_h^{\text{NL}}(x, \tilde{y})) = \tilde{y} = -y \Leftrightarrow \phi_h(x) \leq 2$ (Lemma 2).
 $\phi_h(x) \leq 2 \Leftrightarrow \frac{|w^\top x|}{\|w\|} \leq 2$ (Observation).
 $\frac{|w^\top x|}{\|w\|} \leq 2 \Leftrightarrow \frac{\tilde{y}w^\top x}{\|w\|} \geq -2$ ($\tilde{y}w^\top x < 0$).
 $\frac{\tilde{y}w^\top x}{\|w\|} \geq -2 \Leftrightarrow \frac{yw^\top x}{\|w\|} < -2\tilde{y}y$

□

E.6 Hard formulation equivalent form (Appendix D.1)

PROOF. We begin by rewriting Eq. 30:

$$\underset{w: \|w\|=1}{\operatorname{argmax}} \min_{i \in [m]} y_i(w^\top x_i + 2\tilde{y}_i) \quad (31)$$

The two are equivalent whenever there is a solution to the preceding problem, i.e. the data is strategically separable (Definition 2). Let w^* be a solution of Eq. 31. Define $\gamma^* = \min_{i \in [m]} y_i(w^\top x_i + 2\tilde{y}_i)$. Therefore $\forall i \in [m]$ we have:

$$y_i((w^*)^\top x_i + 2\tilde{y}_i) \geq \gamma^*$$

Or equivalently (since $\gamma^* > 0$ and $\|w^*\| = 1$):

$$y_i\left(\left(\frac{w^*}{\gamma^*}\right)^\top x_i + 2\tilde{y}_i \left\|\frac{w^*}{\gamma^*}\right\|\right) \geq 1$$

Hence, $\frac{w^*}{\gamma^*}$ satisfies the conditions of the optimization in Algorithm 1. Therefore, $\|w_0\| \geq \left\|\frac{w^*}{\gamma^*}\right\| = \frac{1}{\gamma^*}$.

It follows that $\forall i \in [m]$:

$$y_i(\hat{w}^\top x_i + 2\tilde{y}_i) = \frac{1}{\|w_0\|} y_i(w_0^\top x_i + 2\tilde{y}_i \|w_0\|) \geq \frac{1}{\|w_0\|} \geq \gamma^*$$

Since $\|\hat{w}\| = 1$ we obtain that \hat{w} is an optimal solution of Eq. 31.

□

E.7 Soft formulation equivalent form (Appendix D.2)

PROOF. Fix some w, i and consider the minimization over ξ_i in Algorithm 2. Since ξ_i must be nonnegative, the best assignment to x_i would be 0 if $y_i(w^\top x_i + 2\tilde{y}_i \|w\|) \geq 1$ and would be $1 - y_i(w^\top x_i + 2\tilde{y}_i \|w\|)$ otherwise. Therefore

$$\xi_i = \begin{cases} 0, & y_i(w^\top x_i + 2\tilde{y}_i \|w\|) \geq 1 \\ 1 - y_i(w^\top x_i + 2\tilde{y}_i \|w\|), & y_i(w^\top x_i + 2\tilde{y}_i \|w\|) < 1 \end{cases} = \max\{0, 1 - y_i(w^\top x_i + 2\tilde{y}_i \|w\|)\}$$

□

E.8 Definition 4 equivalent form

PROOF. We will show that for the NL setting:

$$L_{\text{s-hinge}}^{\text{NL}}(x, \tilde{y}, y; w) \triangleq \max\{0, 1 - y(w^\top x + 2\tilde{y} \|w\|)\} = \max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) - (2 - c(x, \Delta_h^{\text{NL}}(x, \tilde{y})))y\tilde{y} \|w\|\}$$

Assumption:

Users do not change their features unless the change strictly increase their utility, i.e. $h(\Delta_h^{\text{NL}}(x; \tilde{y})) = h(x) \Rightarrow \Delta_h^{\text{NL}}(x; \tilde{y}) = x$

Observation:

Assuming $c(x, x') = \|x - x'\|$ as in NL, the closest point x' to x for which $h(x') \neq h(x)$ is a point on the hyperplane w closest to x . The modification cost is the euclidean distance between x and the hyperplane ($c(x, x') = \frac{|w^\top x|}{\|w\|}$)

Let $h = (w)$ be a linear classifier and $(x, \tilde{y}, y) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$.

- If $\frac{\tilde{y}w^\top x}{\|w\|} > 0$:
 $\tilde{y}w^\top x > 0 \Rightarrow$
 $h(x) = \tilde{y} \Rightarrow$ (Lemma 2).
 $h(\Delta_h^{\text{NL}}(x; \tilde{y})) = h(x) = \tilde{y} \Rightarrow$ (assumption).
 $\Delta_h^{\text{NL}}(x; \tilde{y}) = x \Rightarrow$
 $yw^\top \Delta_h^{\text{NL}}(x; \tilde{y}) = yw^\top x \Rightarrow$ ($c(x, \Delta_h^{\text{NL}}(x; \tilde{y})) = c(x, x) = 0$)
 $\max\{0, 1 - y(w^\top x + 2\tilde{y}\|w\|)\} = \max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) - (2 - c(x, \Delta_h^{\text{NL}}(x, \tilde{y})))y\tilde{y}\|w\|\}$
- If $0 \geq \frac{\tilde{y}w^\top x}{\|w\|} \geq -2$:
 $\tilde{y}w^\top x < 0 \Rightarrow$
 $h(x) \neq \tilde{y} \Rightarrow$ (Lemma 2).
 $(h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq h(x) \Leftrightarrow \phi_h(x) \leq 2) \Rightarrow$ ($\phi_h(x) = \frac{|w^\top x|}{\|w\|}$ (see observation). Therefore $\phi_h(x) \leq 2$).
 $h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq h(x) \Rightarrow$ (observation).
 $yw^\top \Delta_h^{\text{NL}}(x; \tilde{y}) = 0 \wedge c(x, \Delta_h^{\text{NL}}(x; \tilde{y})) = \frac{|w^\top x|}{\|w\|} \Rightarrow$ ($\frac{|w^\top x|}{\|w\|} = \frac{-\tilde{y}w^\top x}{\|w\|}$).
 $\max\{0, 1 - y(w^\top x + 2\tilde{y}\|w\|)\} = \max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) - (2 - c(x, \Delta_h^{\text{NL}}(x, \tilde{y})))y\tilde{y}\|w\|\}$
- If $\frac{\tilde{y}w^\top x}{\|w\|} < -2$: $\tilde{y}w^\top x < 0 \Rightarrow$
 $h(x) \neq \tilde{y} \Rightarrow$ (Lemma 2).
 $(h(\Delta_h^{\text{NL}}(x; \tilde{y})) \neq h(x) \Leftrightarrow \phi_h(x) \leq 2) \Rightarrow$ ($\phi_h(x) = \frac{|w^\top x|}{\|w\|}$ (see Lemma 1 proof). Therefore $\phi_h(x) > 2$).
 $h(\Delta_h^{\text{NL}}(x; \tilde{y})) = h(x) \Rightarrow$ (assumption).
 $\Delta_h^{\text{NL}}(x; \tilde{y}) = x \Rightarrow$
 $yw^\top \Delta_h^{\text{NL}}(x; \tilde{y}) = yw^\top x \Rightarrow$ ($c(x, \Delta_h^{\text{NL}}(x; \tilde{y})) = c(x, x) = 0$)
 $\max\{0, 1 - y(w^\top x + 2\tilde{y}\|w\|)\} = \max\{0, 1 - yw^\top \Delta_h^{\text{NL}}(x, \tilde{y}) - (2 - c(x, \Delta_h^{\text{NL}}(x, \tilde{y})))y\tilde{y}\|w\|\}$

□

E.9 strategic distance for GP

$$d_\Delta^{\text{GP}}(x, z; w) = |\bar{w}^\top x + 2z|$$

PROOF. Let $h = (w)$ be a linear classifier, $x, x' \in \mathcal{X}$ and $z \in \mathcal{Z}$. We show necessary and sufficient conditions for which $h(\Delta_h^{\text{GP}}(x)) \neq h(\Delta_h^{\text{GP}}(x'))$.

- If $h(x) = z$:
 $h(\Delta_h^{\text{GP}}(x)) \neq h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).
 $z \neq h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).
 $h(x') \neq z \wedge \phi_h(x') > 2 \Leftrightarrow$
 $zw^\top x' < 0 \wedge \frac{|w^\top x'|}{\|w\|} > 2 \Leftrightarrow$
 $\frac{zw^\top x'}{\|w\|} < -2$.
Also, $zw^\top x > 0$. Therefore x and any x' for which $\frac{zw^\top x'}{\|w\|} < -2$ are on different sides of the hyperplane.

The minimal distance from x to such x' is the distance from x to the hyperplane plus 2:

$$d_\Delta^{\text{GP}}(x, z; w) = \frac{|w^\top x|}{\|w\|} + 2 = \frac{zw^\top x}{\|w\|} + 2 = \left| \frac{w^\top x}{\|w\|} + 2z \right| = |\bar{w}^\top x + 2z|$$

- If $h(x) \neq z$:

- If $\phi_h(x) \leq 2$:
 $h(\Delta_h^{\text{GP}}(x)) \neq h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).

$z \neq h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).

$h(x') \neq z \wedge \phi_h(x') > 2 \Leftrightarrow$

$zw^\top x' < 0 \wedge \frac{|w^\top x'|}{\|w\|} > 2 \Leftrightarrow$

$\frac{zw^\top x'}{\|w\|} < -2.$

Also, $zw^\top x \leq 0$ and $\frac{|w^\top x|}{\|w\|} \leq 2$.

This means that $-2 \leq \frac{zw^\top x'}{\|w\|} \leq 0$.

Therefore x and any x' for which $\frac{zw^\top x'}{\|w\|} < -2$ are on the same side of the hyperplane and x is closer to the hyperplane than x' .

The minimal distance from x to such x' is 2 minus the distance from x to the hyperplane:

$$d_\Delta^{\text{GP}}(x, z; w) = 2 - \frac{|w^\top x|}{\|w\|} = 2 + \frac{zw^\top x}{\|w\|} = |\frac{w^\top x}{\|w\|} + 2z| = |\bar{w}^\top x + 2z|$$

If $\phi_h(x) > 2$:

$h(\Delta_h^{\text{GP}}(x)) \neq h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).

$z = h(\Delta_h^{\text{GP}}(x')) \Leftrightarrow$ (Lemma 2).

$h(x') = z \vee (h(x') \neq z \wedge \phi_h(x') \leq 2) \Leftrightarrow$

$zw^\top x' > 0 \vee (zw^\top x' \leq 0 \wedge \frac{|w^\top x'|}{\|w\|} \leq 2) \Leftrightarrow$

$\frac{zw^\top x'}{\|w\|} > 0 \vee (-2 \leq \frac{zw^\top x'}{\|w\|} \leq 0) \Rightarrow$

$\frac{zw^\top x'}{\|w\|} \geq -2.$

Also, $zw^\top x \leq 0$ and $\frac{|w^\top x|}{\|w\|} > 2$.

This means that $\frac{zw^\top x'}{\|w\|} < -2$.

The minimal distance from x to an x' for which $\frac{zw^\top x'}{\|w\|} \geq -2$ is the distance from x to the hyperplane minus 2:

$$d_\Delta^{\text{GP}}(x, z; w) = \frac{|w^\top x|}{\|w\|} - 2 = -\frac{zw^\top x}{\|w\|} - 2 = |\frac{w^\top x}{\|w\|} + 2z| = |\bar{w}^\top x + 2z|.$$

□

E.10 Theorem 1

PROOF. Denote by s a sample $(x, z, y) \in \mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$. Recall:

$$\mathcal{L}_{0/1}^{\text{GSC}}(s; w) \triangleq \mathbb{1}\{y \text{ sign}(w^\top \Delta_h(x; z))\}$$

$$L_{\text{s-hinge}}^{\text{GSC}}(s; w) \triangleq \max\{0, 1 - y \text{ sign}(w^\top \Delta_h(x, z)) d_\Delta(x, z; w) \|w\|\}$$

Since $d_\Delta(x, z; w) \geq 0$ it is clear that $\mathcal{L}_{0/1}^{\text{GSC}}(s; w) \leq L_{\text{s-hinge}}^{\text{GSC}}(s; w)$.

We rewrite $L_{\text{s-hinge}}^{\text{GSC}}(s; w)$:

$$L_{\text{s-hinge}}^{\text{GSC}}(s; w) \triangleq \max\{0, 1 - y \text{ sign}(w^\top \Delta_h(x, z)) \|x - x^d\| \|w\|\}$$

where

$$x^d \triangleq \underset{x'}{\operatorname{argmin}} \|x - x'\|$$

$$\text{s.t. } h(\Delta_h(x, z)) \neq h(\Delta_h(x', z))$$

Define $\mathcal{H}_k = \{s \rightarrow y \text{ sign}(w^\top \Delta_h(x, z)) \|x - x_d\| \|w\| : \|w\| \leq k\}$ and let $S = \{s_i\}^m$ be vectors in that space. Denote $r = \max_{x \in \mathcal{X}} \|x\|$. We bound the Rademacher complexity of $\mathcal{H}_k \circ S$:

$$\begin{aligned}
mR(\mathcal{H}_k \circ S) &= \mathbb{E}_{\sigma} \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i y_i \operatorname{sign}(w^\top \Delta_h(x_i, z_i)) \|x_i - x_i^d\| \|w\| \right] =_1 \\
&\mathbb{E}_{\sigma} \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i \|x_i - x_i^d\| \|w\| \right] = \mathbb{E}_{\sigma} \left[\sup_{w \in \mathcal{H}_k} \|w\| \sum_{i=1}^m \sigma_i \|x_i - x_i^d\| \right] \leq \\
&\mathbb{E}_{\sigma} \left[k \left| \sum_{i=1}^m \sigma_i \|x_i - x_i^d\| \right| \right] = k \mathbb{E}_{\sigma} \left[\sqrt{\left(\sum_{i=1}^m \sigma_i \|x_i - x_i^d\| \right)^2} \right] \leq_2 \\
&k \sqrt{\mathbb{E}_{\sigma} \left[\left(\sum_{i=1}^m \sigma_i \|x_i - x_i^d\| \right)^2 \right]} = k \sqrt{\mathbb{E}_{\sigma} \left[\sum_{i,j} \sigma_i \sigma_j \|x_i - x_i^d\| \|x_j - x_j^d\| \right]} =_3 \\
&k \sqrt{\sum_{i=1}^m \mathbb{E}_{\sigma} [\sigma_i^2] \|x_i - x_i^d\|^2} \leq k \sqrt{m} \cdot \max_i \|x_i - x_i^d\| \leq k \sqrt{m} \cdot \max_i \|x_i\| + \|x_i^d\| \leq k \sqrt{m} \cdot 2r
\end{aligned}$$

1. $\sigma_i y_i \operatorname{sign}(w^\top \Delta_h(x_i, z_i)) \equiv_d \sigma_i$ under the expectancy

2. Jensen's inequality

3. Independence

Therefore $R(\mathcal{H}_k \circ S) \leq \frac{2kr}{\sqrt{m}}$.

Consider the hinge function $\phi(t) = \max\{0, 1-t\}$ for scalar values $t \in \mathbb{R}$. ϕ is 1-lipschitz:

$$\forall t_1, t_2 \in \mathbb{R}$$

$$|\phi(t_1) - \phi(t_2)| = |\max 0, 1-t_1 - \max 0, 1-t_2| \leq \max |0-0|, |1-t_1-1+t_2| = |t_1 - t_2|$$

Then, according to Talagrand's contraction principal, $R(\phi \circ \mathcal{H}_k) \leq R(\mathcal{H}_k)$.

Let D be a distribution on $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$ such that there exists some w^* with $\mathbb{P}_D [y \operatorname{sign}((w^*)^\top \Delta_h(x, z) \geq 1] = 1$. Let w_S be the output of the Hard formulation algorithm without normalization:

$$\begin{aligned}
w &= \underset{w}{\operatorname{argmin}} \|w\|^2 \\
\text{s.t. } \forall i \in [m], y_i \operatorname{sign}(w^\top \Delta_h(x_i, z_i)) \|x_i - x_i^d\| \|w\| &\geq 1
\end{aligned}$$

Define $\mathcal{H}^* = \{w : \|w\| \leq \|w^*\|\}$.

$\|w_S\| \leq \|w^*\|$ and therefore $w_S \in \mathcal{H}^*$.

From the generalization theorem on Rademacher complexity, with probability greater or equal to $1 - \delta$, for all $w \in \mathcal{H}^*$:

$$\mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; D) - \mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; S) \leq 2R_D(\phi \circ \mathcal{H}_{\|w^*\|}) + c \sqrt{\frac{2 \ln(\frac{2}{\delta})}{m}}$$

where c is the maximal loss, which in our case is $(1 + 2r \|w^*\|)$.

Finally, we get:

$$\mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; D) \leq \mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; S) + \frac{4r \|w^*\|}{\sqrt{m}} + (1 + 2r \|w^*\|) \sqrt{\frac{2 \ln(\frac{2}{\delta})}{m}}$$

Notice that w^* is unknown. Therefore we show a data-dependant bound:

Define $\mathcal{H}^i = \{w : \|w\| \leq 2^i\}$ and $\delta_i = \frac{\delta}{2^i}$. Note that $\sum_{i=1}^{\infty} \delta_i = \delta$.

Similarly to the first part of the proof, for all i , $\forall w \in \mathcal{H}_i$ with probability greater or equal to δ :

$$\mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; D) \leq \mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; S) + \frac{4r \cdot 2^i}{\sqrt{m}} + (1 + 2 \cdot 2^i) \sqrt{\frac{2 \ln(\frac{2}{\delta_i})}{m}}$$

From the union bound, we get that with probability greater or equal to $1 - \delta$ this holds for all \mathcal{H}_i . This means that for all $w \in \mathcal{H}$ we have for $i = \lceil \log(\|w\|) \rceil \leq \log(\|w\|) + 1$:

$$\mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; D) \leq \mathcal{L}_{\text{s-hinge}}^{\text{GSC}}(w; S) + \frac{8r \|w\|}{\sqrt{m}} + (1 + 4r \|w\|) \sqrt{\frac{2 \ln(\frac{4\|w\|}{\delta})}{m}}$$

Plugging $w = w_S$ finishes the proof. This proof can be adjusted easily to work for the soft formulation algorithm. \square

E.11 Tightened generalization bounds for notable subclasses of GSC

Let D be a joint distribution over triples (x, z, y) , and let $r = \max_{x \in X} \|x\|$. Denote by \hat{w} the minimizer of the empirical generalized strategic hinge loss with L_2 regularization. Then for every $\delta \in [0, 1]$, if the training set S includes m samples, then w.p. $\geq 1 - \delta$ it holds that:

$$\mathcal{L}_{0/1} \leq \hat{\mathcal{L}}_{\text{s-hinge}}^{\text{GP}} + \frac{4r \|\hat{w}\|}{\sqrt{m}} + (1 + 2\rho^{\text{GP}} \|w\|) \sqrt{\frac{2 \ln(\frac{4\|\hat{w}\|}{\delta})}{m}}$$

where $\mathcal{L}_{0/1}$ is the expected 0/1 loss, $\hat{\mathcal{L}}_{\text{s-hinge}}^{\text{GP}}$ is the empirical s-hinge loss on S , and $\rho^{\text{GP}} = r + 2$.

PROOF. This proof follows the proof of Theorem E.11 with two slight differences:

- Rademacher complexity of $\mathcal{H}_k \circ S$:

$$\begin{aligned} mR(\mathcal{H}_k \circ S) &= \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i y_i (w^\top x_i + 2z_i \|w\|) \right] = 1 \\ \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i (w^\top x_i + 2z_i \|w\|) \right] &= \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i (w^\top x_i + 2z_i \|w\|) \right] \leq \\ \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i w^\top x_i \right] + \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} \sum_{i=1}^m \sigma_i \cdot 2z_i \|w\| \right] &= 2 \\ \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} w^\top \sum_{i=1}^m \sigma_i x_i \right] + \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} 2\|w\| \sum_{i=1}^m \sigma_i \right] &= \\ \mathbb{E}_\sigma \left[\sup_{w \in \mathcal{H}_k} w^\top \sum_{i=1}^m \sigma_i x_i \right] &\leq_3 \mathbb{E}_\sigma \left[k \left\| \sum_{i=1}^m \sigma_i x_i \right\| \right] = \\ k \mathbb{E}_\sigma \left[\sqrt{\left\| \sum_{i=1}^m \sigma_i x_i \right\|^2} \right] &\leq_4 k \sqrt{\mathbb{E}_\sigma \left[\left\| \sum_{i=1}^m \sigma_i x_i \right\|^2 \right]} = \\ k \sqrt{\mathbb{E}_\sigma \left[\sum_{i,j}^m \sigma_i \sigma_j \langle x_i, x_j \rangle \right]} &= 5 k \sqrt{\sum_{i=1}^m \mathbb{E}_\sigma [\sigma_i^2] \|x_i\|^2} \leq \\ k \sqrt{m} \cdot \max_i \|x_i\| &\leq k \sqrt{m} \cdot \max_i \|x_i\| = k \sqrt{m} \cdot r \end{aligned}$$

1. $\sigma_i y_i \equiv_d \sigma_i$ under the expectancy
2. $\sigma_i z_i \equiv_d \sigma_i$ under the expectancy
3. Cauchy-Schwartz inequality
4. Jensen's inequality
5. Independence

Therefore $R(\mathcal{H}_k \circ S) \leq \frac{kr}{\sqrt{m}}$.

- $\rho^{\text{NL}}, \rho^{\text{GP}}, \rho^{\text{ADV}}, \rho^{\text{SC}}$:

The Rademacher complexity generalization theorem states that c is equal to the maximal loss for any sample in S . Therefore for GP, ADV and SC:

$$\begin{aligned} \max_{s \in S} \mathcal{L}_{\text{s-hinge}}(s) &= \max_{s \in S} \max \{0, 1 - y w^\top x + 2yz \|w\|\} = 1 + (r+2) \|w\| \Rightarrow \\ &\Rightarrow \rho^{\text{GP}}, \rho^{\text{ADV}}, \rho^{\text{SC}} = r+2 \end{aligned}$$

However, a close examination of the theorem proof [10] allows for the partition of the sample-set S into several subsets S_1, S_2, \dots, S_k . Then $c = \sum_{i=1}^k \frac{\mathbb{E}_D[|S_i|]}{|S|} \max_{s \in S_i} \text{loss}(s)$.

In NL, the sample set can be partitioned into 2 subsets:

$$S_1 = \{s = (x, \tilde{y}, y) \in S : \tilde{y} = y\}, S_2 = \{s = (x, \tilde{y}, y) \in S : \tilde{y} \neq y\}.$$

$$\mathbb{E}_D[|S_1|] = 1 - \epsilon, \mathbb{E}_D[|S_2|] = \epsilon.$$

$$\max_{s \in S_1} \mathcal{L}_{\text{s-hinge}}^{\text{NL}}(s) = \max_{s \in S_1} \max\{0, 1 - yw^\top x - 2\|w\|\} = 1 + (r - 2)\|w\|.$$

$$\max_{s \in S_2} \mathcal{L}_{\text{s-hinge}}^{\text{NL}}(s) = \max_{s \in S_2} \max\{0, 1 - yw^\top x + 2\|w\|\} = 1 + (r + 2)\|w\|.$$

$$\text{In conclusion, } c = \epsilon(1 + (r + 2)\|w\|) + (1 - \epsilon)(1 + (r - 2)\|w\|) = 1 + (r + 2 - 4\epsilon)\|w\|.$$

□