Improved Robust Algorithms
for Learning with Discriminative Feature Feedback

Sivan Sabato
Department of Computer Science
Ben-Gurion University of the Negev

Abstract

Discriminative Feature Feedback is a setting first introduced by Dasgupta et al. (2018), which provides a protocol for interactive learning based on feature explanations that are provided by a human teacher. The features distinguish between the labels of pairs of possibly similar instances. That work has shown that learning in this model can have considerable statistical and computational advantages over learning in standard label-based interactive learning models.

In this work, we provide new robust interactive learning algorithms for the Discriminative Feature Feedback model, with mistake bounds that are significantly lower than those of previous robust algorithms for this setting. In the adversarial setting, we reduce the dependence on the number of protocol exceptions from quadratic to linear. In addition, we provide an algorithm for a slightly more restricted model, which obtains an even smaller mistake bound for large models with many exceptions. In the stochastic setting, we provide the first algorithm that converges to the exception rate with a polynomial sample complexity. Our algorithm and analysis for the stochastic setting involve a new construction that we call *Feature Influence*, which may be of wider applicability.

1 INTRODUCTION

Interactive Machine Learning is an umbrella term for settings in which the learning algorithm interactively obtains feedback from the environment while learning. Such settings, and especially Active Learning, have drawn much

interest in recent years, as interactivity can significantly improve the results of learning (see, e.g., Hanneke et al., 2014). Incorporating human feedback beyond labels into the interactive process has a great potential for further improvements, as it can allow learning with less data, and can result in more accurate models that are also more interpretable.

We consider *Discriminative Feature Feedback*, an interactive learning model first introduced by Dasgupta et al. (2018). This model provides a formal framework for the idea that human teachers can in many cases provide *explanations* for the reasons why certain examples should be classified in a specific way, based on a comparison between examples. For instance, when classifying patients according to their underlying condition, an expert may explain that patient A does not have the same condition as patient B, even though their parameters may seem similar, because patient A has a blood test result that is inconsistent with the condition of patient B. Dasgupta et al. (2018) showed that this richer type of feedback can make learning significantly easier than standard label-based approaches. An important property of the Discriminative Feature Feedback model is that the a-priori number of possible features (the dimensionality of the problem) need not be bounded. Indeed, the provided mistake bounds are independent of the number of possible features.

The model proposed in Dasgupta et al. (2018) was nonetheless limited, as it required the feedback from the human teacher to always conform to the Discriminative Feature Feedback protocol. Deviations from the protocol can occur because of human error, as well as because of inherent exceptional behavior of certain examples, such as patients who have a rare condition that is difficult to identify using available measures. Dasgupta and Sabato (2020) showed that handling such deviations effectively requires relaxing the protocol assumptions of the original model, and considered a setting in which a bounded amount (number or rate) of exceptions from the feature feedback protocol is allowed. Under this relaxation, they provided robust algorithms for the adversarial setting and for the stochastic setting with dimension-independent mistake bounds. However, the mistake bounds derived in Dasgupta and Sabato (2020) depend

very strongly on the exception threshold: In the adversarial setting, the mistake bound is quadratic in the number of exceptions, and in the stochastic setting, whenever the exception rate is non-zero, the mistake rate of the algorithm does not converge to the exception rate.

In this work, we provide new algorithms for the adversarial setting and for the stochastic setting, which obtain significantly improved mistake bounds. For the adversarial setting, we achieve a linear dependence on the number of exceptions, and in the stochastic case we show that it is possible to converge to the exception rate, with a polynomial sample complexity. We also show that in the adversarial setting, a slightly more restricted feature feedback model leads to a significantly improved mistake bound. Lastly, we show that the algorithms can be made parameter-free while keeping the mistake bound the same up to logarithmic factors.

Our algorithm and analysis for the stochastic setting employ a new construction that we call *Feature Influence*, which may be of wider applicability. This construction allows separating the cost of identifying candidate features from the cost of constructing a low-error classification model over these features, and provides a generic recipe for constructing learning algorithms for a wide range of interactive settings.

Paper structure   We discuss related work in Sec. 2. Sec. 3 provides the formal definition of the setting and specific previous results. Sec. 4 presents robust algorithms for the agnostic setting. Sec. 5 provides algorithms for the stochastic setting. We conclude in Sec. 6. Some proofs are deferred to Appendices.

2   RELATED WORK

Several previous works studied explanation-based interactive learning in various application domains. Teso and Kersting (2019) use interactive local machine-generated explanations with user-based corrections to guide interactive learning. Schramowski et al. (2020) show how deep neural network learning can be improved using interactive explanations. Guo et al. (2022) studied the effect of explanation-based interactive learning on learning outcomes.

Other works specifically suggest using some form of a feature feedback to improve learning results. (Croft and Das, 1990) studied the use of weakly predictive features, and variations of this idea were studied in various applications such as text and vision (Raghavan et al., 2005; Druck et al., 2008; Settles, 2011; Mac Aodha et al., 2018; Liang et al., 2020). Poulis and Dasgupta (2017); Visotsky et al. (2019) studied formal versions of these approaches and their impact on the sample complexity.

Interactive learning with user feedback that provides features distinguishing between instances have been used in computer vision (Branson et al., 2010; Zou et al., 2015).

The formal Discriminative Feature Feedback setting was first introduced in Dasgupta et al. (2018), and the first robust algorithms for this setting were proposed in Dasgupta and Sabato (2020). In the next section, we provide a formal definition of the setting and discuss relevant previous results.

3   SETTING AND PRELIMINARIES

The formal definition of the Discriminative Feature Feedback model provided below follows Dasgupta et al. (2018) and Dasgupta and Sabato (2020). Let $\mathcal{X}$ be the domain of examples, and let $\mathcal{Y}$ be a finite domain of labels. Let $c^* : \mathcal{X} \to \mathcal{Y}$ be the unknown target concept mapping examples to labels. Let Φ be a set of binary features, where each feature $\phi \in \Phi$ is some predicate $\phi : \mathcal{X} \to \{\texttt{true}, \texttt{false}\}$. Φ can be infinite, and is not necessarily known to the learner. Φ is assumed to be closed under negation.

The Discriminative Feature Feedback protocol is defined with respect to a *component representation* which is unknown to the learner: For an integer m, let $\mathcal{G} = \{G_1, \ldots, G_m\}$ be a set of m subsets of $\mathcal{X}$ (components) that cover the domain, $\mathcal{X} = \cup_{i \in [m]} G_i$. For $x \in \mathcal{X}$, let $G(x)$ be some $G \in \mathcal{G}$ such that $x \in G$. It is assumed that each $G \in \mathcal{G}$ has an associated label $\ell(G) \in \mathcal{Y}$, and for each $x \in \mathcal{X}$, $c^*(x) = \ell(G(x))$. Note that components with the same label may overlap.

The central assumption of the Discriminative Feature Feedback model is that for any two G_i, G_j such that $\ell(G_i) \neq \ell(G_j)$, there exists a *discriminative feature* $\phi(G_i, G_j)$, such that x satisfies $\phi(G_i, G_j)$ if $x \in G_i$, and x does not satisfy $\phi(G_i, G_j)$ if $x \in G_j$. Note that there could be more than one potential discriminative feature between two components. We denote by $\phi(\cdot, \cdot)$ a specific choice of such features that satisfies $\forall i, j, \phi(G_j, G_i) = \neg \phi(G_i, G_j)$.

We first describe the feature feedback interactive learning protocol in the pure, noiseless model. Each iteration of the protocol is of the following form:

- A new instance x_t arrives.

- The learning algorithm provides a predicted label $\widehat{y}_t$, and an instance $\widehat{x}_t$ which was previously observed with that label. This instance serves as the explanation for the predicted label: "x_t is assigned label $\hat{y}_t$ because $\hat{x}_t$ was assigned label $\hat{y}_t$".

- If the prediction is correct, no feedback is obtained from the teacher.

- If the prediction is incorrect, the teacher provides the correct label $y_t = c^*(x)$, and the discriminative feature $\phi(G(x_t), G(\widehat{x}_t))$. This feature explains why x_t should be labeled differently from $\widehat{x}_t$.

For example, if the instances represent animal species and the labels represent the type of each animal, then a discriminative feature ϕ between a dolphin and a halibut could be "has gills", explaining why a dolphin is not a fish.

Dasgupta and Sabato (2020) relaxed the above model to allow deviations from the protocol, and studied the adversarial setting and the stochastic setting. In the adversarial setting, the input stream is arbitrary, and it is assumed that there exists a representation of size m such that the interaction with the teacher conforms to the protocol above, except for at most k exceptions. Here, an exception can mean that the teacher provides a label that is inconsistent with the representation or a discriminative feature that is inconsistent with the representation, or both. In the stochastic setting, it is assumed that the input stream is drawn i.i.d. from some distribution over $\mathcal{X}$, and that there exists a representation of size m such that the interaction with the teacher conforms to the protocol above except for a rate of at most ϵ of exceptions. Note that in both cases, there can be more than one pair of representation size (m) and exception threshold (k or ϵ) that are consistent with the input; The bounds discussed below hold for all such pairs.

We use standard $O(\cdot)$ notation to indicates implicit constant factors, and $\tilde{O}(\cdot)$ to indicate implicit logarithmic factors. Dasgupta et al. (2018) provided an algorithm (henceforth denoted DFF18) for the adversarial setting with no exceptions, with a mistake bound of $O(m^2)$. It was shown in Dasgupta and Sabato (2020) that this is the best possible dependence on m. Dasgupta and Sabato (2020) further showed that describing a representation of size m with k exceptions using a different representation with zero exceptions may require $\Omega(|\Phi|)$ components. Thus, to obtain mistake bounds that are independent of $|\Phi|$, robust algorithms that can handle exceptions directly are crucial. Recall that Φ can be infinite, thus bounds that depend on $|\Phi|$ are undesirable.

Dasgupta and Sabato (2020) provided robust algorithms for the adversarial setting and for the stochastic setting with mistake bounds of $O(m^2 k + mk^2)$ and $\tilde{O}(m^2 \epsilon + m^2/\sqrt{n})$ respectively, where n is the length of the stochastic input stream. These mistake bounds incur a substantial penalty for each exception. In this work, we derive new algorithms that obtain significantly improved mistake bounds. In particular, in the adversarial setting we prove a mistake bound of $m(m-1) + mk$, giving a linear dependence on the number of exceptions instead of a quadratic dependence, and in the stochastic case we show that it is possible for the mistake rate to converge exactly to the exception rate with a polynomial sample complexity, unlike the previous bound that only converges to m^2 times the exception rate.

4 ALGORITHMS FOR THE ADVERSARIAL SETTING

In this section, we consider the adversarial setting. In this setting, the input stream can be arbitrary, and it is parameterized only by the representation size m, and by the number of exceptions from the feature-feedback protocol with respect to this representation, which we assume is upper bounded by an integer k. In Sec. 4.1, we derive a robust algorithm for this setting with a mistake bound that avoids the quadratic dependence on k of the algorithm proposed in Dasgupta and Sabato (2020). The mistake bound is proved in Sec. 4.2. In Sec. 4.3, we study a more restricted representation model that assumes a unique label for each component, and show that under this model it is possible to obtain a mistake bound of $2m(m-1) + 6k$. This is a substantially lower dependence on the number of exceptions for large models. This result pinpoints the challenge of improving the mk term in the unrestricted representation model to a specific property of this model: the sharing of labels between components.

4.1 An Improved Robust Algorithm for the Adversarial Setting

We present a new robust algorithm for the adversarial setting with an improved mistake bound. Surprisingly, unlike the algorithm proposed in Dasgupta and Sabato (2020), this new algorithm is only slightly different from the non-robust DFF18. SR-DFF (Simple Robust DFF) is listed in Alg. 1. Like DFF18, SR-DFF keeps track of the following information: The first labeled example (x_0, y_0); A decision list L, which is represented as a list of prediction rules; Each rule in L has a conjunction $C[x]$ of features from Φ, where x is some example from $\mathcal{X}$ that represents the rule, and a label associated with the rule, denoted $\text{label}[x]$. The conjunctions in the list are iteratively refined based on the feedback from the teacher. A rule is *created* if an example that matches none of the existing conjunctions appears. A rule is *refined* if mistakes occur on an example that matched the rule. A rule is *deleted* if it becomes too long. As we show below, a rule can become too long if it was refined using feedback based on exceptions, or if the rule itself was created based on an exception. The option to delete a rule is the main change compared to DFF18.

We derive the following mistake bound for SR-DFF.

Theorem 1. *Suppose that the input stream is consistent with some representation of size m with at most k exceptions. Then SR-DFF makes at most $m(m-1) + mk$ mistakes.*

This mistake bound is a factor of k smaller compared to the mistake bound of $O(m^2 k + mk^2)$ of the robust algorithm of Dasgupta and Sabato (2020). Unlike the non-robust algorithm DFF18, SR-DFF requires the maximal number of components m as input. The robust algorithm of Dasgupta and Sabato (2020) required both m and k as input, thus this

Algorithm 1 `SR-DFF`: A simple robust Discriminative Feature Feedback algorithm for the adversarial setting

Input: Maximal number of components m.
1: $t \leftarrow 0$; Get the label y_0 of the first example x_0.
2: Initialize L to an empty list.
3: **while** `true` **do**
4:     $t \leftarrow t + 1$; Get next example x_t.
5:     **if** $\exists C[\widehat{x}] \in L$ such that x_t satisfies $C[\widehat{x}]$ **then**
6:         Predict `label`$[\widehat{x}]$ and provide explanation $\widehat{x}$.
7:         **if** prediction is incorrect **then**
8:             Get correct label y_t and feature ϕ.
9:             $C[\widehat{x}] := C[\widehat{x}] \wedge \neg\phi$
10:            **if** $|C[\widehat{x}]| \geq m$ **then**
11:                Delete $C[\widehat{x}]$ from L.
12:     **else** (no relevant rule exists)
13:         Predict y_0 and provide explanation x_0.
14:         **if** prediction is incorrect **then**
15:             Get correct label y_t and feature ϕ.
16:             Add to L an empty conjunction $C[x_t]$
17:             Set `label`$[x_t] \leftarrow y_t$.

is an improvement for robust algorithms. Moreover, the following theorem shows that the requirement to have m as input can also be relaxed, while keeping the order of the mistake bound the same up to logarithmic factors.

Theorem 2. *Suppose that the input stream is consistent with some representation of size m with at most k exceptions. Let* $\mathrm{UB}(m, k) := m(m-1) + mk$ *be the mistake upper bound proved in Theorem 1. There exists a parameter-free robust algorithm with a mistake bound of*

$$32\mathrm{UB}(m, k) \log^2(8\mathrm{UB}(m, k)) = \tilde{O}(m^2 + mk).$$

This theorem is proved in App. A. We now turn to prove the main mistake bound, Theorem 1.

4.2 Bounding the Mistakes of `SR-DFF`

We now prove Theorem 1 that bounds the number of mistakes made by `SR-DFF`. Assume some representation $\mathcal{G}$ of size m such that the input has at most k exceptions with respect to this representation. We call a rule maintained by the algorithm *corrupted* if at least one of the features in its conjunction was added in line 9 when the example x_t was an exception. We say that a rule $C[x]$ was created by an exception if x is an exception. Note that a rule can be created by an exception and still not be considered corrupted. Call non-exception examples *valid examples*. Call rules created by valid examples *valid rules*.

We start by proving several invariants of the algorithm. First, we prove that valid non-corrupted rules are always satisfied by a whole component in the representation $\mathcal{G}$.

Lemma 3. *At all times in the algorithm, if $C[\widehat{x}]$ is a valid and non-corrupted rule, then: $C[\widehat{x}]$ is satisfied by every*

example in $G(\widehat{x})$, and for every feature ϕ in $C[\widehat{x}]$, there is some valid example $x \in \mathcal{X}$ such that $\phi(G(\widehat{x}), G(x)) = \phi$.

Proof. We prove the claim by induction on the length of $C[\widehat{x}]$. When $C[\widehat{x}]$ is first created, it is an empty conjunction so it is satisfied by all of $G(\widehat{x})$. When $C[\widehat{x}]$ is refined by $\neg\phi$, since the rule is not corrupted, the example x for which ϕ was provided is valid, hence $\neg\phi = \phi(G(\widehat{x}), G(x))$. This implies that $G(\widehat{x})$ has no examples that are satisfied by ϕ. Hence, after adding $\neg\phi$ to $C[\widehat{x}]$, the refined $C[\widehat{x}]$ is still satisfied by $G(\widehat{x})$ and is separated by ϕ from $G(x)$. $\square$

Next, we prove that two non-corrupted valid rules have representatives from different components.

Lemma 4. *For any two valid non-corrupted rules $C[x]$ and $C[x']$ that exist at the same time in L, $G(x) \neq G(x')$.*

Proof. Suppose x was observed earlier in the input sequence and x' was observed later; Since $C[x]$ and $C[x']$ were both created, this means that $C[x]$, in its form when x' was observed, was not satisfied by x'. But by Lemma 3, $C[x]$ is always satisfied by $G(x)$. Hence, $x' \notin G(x)$, which implies the claim. $\square$

Next, only rules affected by exceptions might be deleted.

Lemma 5. *If the rule $C[\widehat{x}]$ is deleted then at least one of the two holds: $\widehat{x}$ is an exception or $C[\widehat{x}]$ is corrupted.*

Proof. Assume for contradiction that $\widehat{x}$ is not an exception and $C[\widehat{x}]$ is not corrupted, but rule $C[\widehat{x}]$ is deleted. Thus, the conjunction $C[\widehat{x}]$ has at least m features. By Lemma 3, for each feature in $C[\widehat{x}]$ there is some valid example x such that $G(x)$ is separated from $G(\widehat{x})$ using that feature. Moreover, after $C[\widehat{x}]$ is refined with this feature, no additional examples from $G(x)$ will be satisfied by this rule. Thus, each component can contribute at most one feature to $C[\widehat{x}]$. Since there are m components in $\mathcal{G}$, there are at most $m - 1$ features in $C[\widehat{x}]$, a contradiction to the size of $C[\widehat{x}]$. $\square$

We bound the total number of mistakes by first bounding the total number of rules created by the algorithm.

Lemma 6. *`SR-DFF` creates at most $m + k$ rules.*

Proof. By Lemma 4, at any time in the algorithm, the total number of valid non-corrupted rules is at most the number of components, m. Moreover, by Lemma 5, such rules are never deleted. Therefore, the total number of such rules that are generated by the algorithm is at most m. In addition, any one exception cannot both generate a new rule and corrupt an existing one. Therefore, the total number of corrupted rules and rules generated by exceptions over the entire run of the algorithm is at most k. Thus, the total number of rules created by the algorithm is at most $m + k$. $\square$

The proof of Theorem 1 is now immediate: Any rule that makes m mistakes is of length m and is thus deleted. Therefore, each rule makes at most $m - 1$ mistakes, except for rules that end up being deleted, which make m mistakes. By Lemma 6, at most $m + k$ rules are created by the algorithm, and by Lemma 5, at most k of them are deleted. Thus, the total number of mistakes made by the algorithm is at most $m(m - 1) + mk$. This completes the proof of Theorem 1.

4.3 An Improved Mistake Bound for a Restricted Model

According to the mistake bound in Theorem 1, every additional exception can result in as many as m additional mistakes. Is there an algorithm that pays only a constant number of mistakes for every additional exception in the adversarial setting? We now show that such an algorithm does exist for a more restricted representation model. It remains open whether a similar result can be obtained for the general model. In the restricted model, each component has a unique label.

Theorem 7. *Suppose that the input stream is consistent with some representation of size m with at most k exceptions. Suppose further that each component has a unique label. Then there exists a (randomized) algorithm with an expected number of mistakes of at most $2m(m - 1) + 6k$.*

The uniqueness of the label is used by the algorithm to identify which rule needs correction in the case of a mistake. This pinpoints the challenge for the general representation model, and may help in the future to obtain a similar result for that model, or prove its impossibility. The algorithm is listed in Alg. 2. It accepts parameters $p \in [0, 1]$ and $l \in \mathbb{N}$. Theorem 7 is proved using $l = m - 1$ and $p = 1/(m - 1)$.

Similarly to SR-DFF, Alg. 2 refines rules based on feedback on incorrect labels. In addition to the list of rules, Alg. 2 also maintains a counter U for each rule, of the number of updates that have been applied to the rule.[1] When U becomes too large, the rule is deleted. An important difference from SR-DFF is in the behavior of the algorithm when no rule is consistent with the example. In this case, when getting the true label of the example, if no rule with this label already exists, the algorithm creates one with an independent probability of p. If such a rule does exist, then one of the features in its conjunction that disagree with the current example is removed; Note that such a feature must exist, otherwise the rule would be consistent with the example.

The deletion mechanism allows the algorithm to correct corrupted rules using subsequent examples. Several key ideas are incorporated into the algorithm, and lead to a mistake bound with only a constant factor over k:

---

[1]For simplicity, we include in the count of updates also cases where an updated was avoided due to inconsistency (line 9 in Alg. 2) and term these also "updates" below.

Algorithm 2 Robust Discriminative Feature Feedback for the adversarial setting under the unique-label assumption

Input: $p \in [0, 1], l \in \mathbb{N}$
1: $t \leftarrow 0$; Get the label y_0 of the first example x_0.
2: Initialize L to an empty list.
3: **while** true **do**
4:      $t \leftarrow t + 1$; Get next example x_t.
5:      **if** $\exists C[\widehat{x}] \in L$ such that x_t satisfies $C[\widehat{x}]$ **then**
6:          Predict label$[\widehat{x}]$ and provide example $\widehat{x}$.
7:          **if** prediction is incorrect **then**
8:              Get correct label y_t and feature ϕ_t.
9:              **if** $\hat{x}$ satisfies $\neg\phi_t$ and x_t satisfies ϕ_t **then**
10:                  Add $\neg\phi_t$ to the conjunction $C[\hat{x}]$.
11:              $U(\widehat{x}) \leftarrow U(\widehat{x}) + 1$.
12:              **if** $U(\widehat{x}) \geq m + l - 1$ **then** delete rule $C[\widehat{x}]$.
13:      **else**   △ *No rule is satisfied by x_t*
14:          Predict y_0 and provide example x_0.
15:          **if** prediction is incorrect **then**
16:              Get correct label y_t and feature ϕ_t.
17:              **if** there is no rule in L with label y_t **then**
18:                  Draw independently $B \sim \text{Bernoulli}(p)$.
19:                  **if** B = 1 **then**
20:                      Add to L a new empty rule $C[x_t]$;
21:                      Set label$[x_t] = y_t$.
22:                      $U(x_t) \leftarrow 0$.
23:              **else**
24:                  Let $C[\widehat{x}]$ be the rule in L with label y_t.
25:                  Remove from conjunction $C[\widehat{x}]$ some feature that is not satisfied by x_t.
26:                  $U(\widehat{x}) \leftarrow U(\widehat{x}) + 1$.
27:                  **if** $U(\widehat{x}) \geq m + l - 1$ **then**
28:                      delete rule $C[\widehat{x}]$.

- A low probability of creating a new rule ensures that exceptions do not cause the creation of too many rules.

- Removing single features from rules limits the effect of exceptions on the list of rules. This is possible due to the unique-label assumption.

- Rules are deleted only if they are quite long, so that each corrupted rule can absorb several exceptions.

Overall, no single mistake can cause too large a change to the list of rules; This stability property keeps the dependence of the bound on the number of exceptions low.

The full proof of Theorem 7 is provided in App. B. We give here a short proof sketch. First, note that due to the unique-label assumption, for any valid rule $C[\widehat{x}]$ with label y, any valid x_t with the same label must be in $G(\widehat{x})$. Therefore, every time a feature is deleted from a rule in line 25, if x_t and $\widehat{x}$ are both valid examples, removing a feature that is inconsistent with x_t from $C[\widehat{x}]$ is necessarily a correct deletion, in that it removes a feature that does not separate

$G(\widehat{x})$ from any other component. Therefore, the feature must have been added by an exception.

The proof of Theorem 7 bounds the number of total mistakes made by the algorithm using the sum of the number of exceptions and the possible outcomes of a mistake on a valid example. The number of updates to valid rules by valid examples is bounded by observing that all such updates must either add a true separating feature to a component or delete a feature that was added by an exception. The number of valid rules created during the run of the algorithm is bounded by observing that there can never be more rules than components at any time in the run, and by showing that each deleted rule must have been used for predicting at least $l/2$ exceptions. This can be concluded from the following lemma, proved in App. B:

Lemma 8. *Let t be some iteration of the run of Alg. 2. Let $C[x]$ be some valid rule that exists during this iteration, and let $U(x)$ be the value of the update counter for this rule at the end of this iteration. The number of exceptions that caused updates to this rule until the end of this iteration is at least $(U(x) - (m-1))/2$.*

The proof of Theorem 7 further uses the probability p of success in creating a new rule to bound to the expected total number of rules created. This, along with the deletion of rules that have had too many updates, allows bounding the total number of mistakes caused by rules with exceptions as representatives. The full analysis is provided in App. B.

5 THE STOCHASTIC SETTING — APPROACHING THE OPTIMAL MISTAKE RATE

In this section, we assume that the input stream of examples is drawn from a stochastic source, which outputs i.i.d. examples according to a distribution. It is assumed that the distribution and the provided teacher feedback are consistent with some representation $\mathcal{G}$ of size m and its induced concept c^*, except for a rate of up to $\epsilon \in [0,1)$ of exceptions.

Formally, there exists some marginal distribution $\mathcal{D}_X$ over examples from $\mathcal{X}$. Examples are drawn i.i.d. according to $\mathcal{D}_X$. For each draw of $X \sim \mathcal{D}_X$, there is some probability that an exception occurs, which can depend on the value of X. The overall probability of an exception is at most ϵ. If an exception does not occur, then the label feedback and feature feedback, if provided, are consistent with the representation and the feedback protocol. Letting $\mathcal{Y}$ be the set of possible labels, we denote by $\mathcal{D}$ the distribution over $\mathcal{X} \times \mathcal{Y}$ which is induced by drawing an example $X \sim \mathcal{D}_X$ and assigning the label $Y = c^*(X)$ if there is no exception, and some example and label induced by the exception otherwise.[2] The labeled

---

[2]In fact, in case of an exceptions the example and label can be adversary and do not need to be random. We adhere to the distribution formulation for simplicity of presentation.

examples provided to the learner in the stochastic setting are distributed as i.i.d. draws from $\mathcal{D}$.

The algorithms for the adversarial settings provided in Sec. 4, as well as their mistake bounds, apply also in a stochastic setting. However, since $\mathbb{E}[k] = \epsilon n$, Theorem 1 gives a mistake rate of $m(m-1)/n + \epsilon m$, so that even when $n \to \infty$, the mistake rate of SR-DFF never converges to the exception rate, but to a factor of m over it. The bound for the stochastic setting in Dasgupta and Sabato (2020) gives an even larger factor of m^2. Theorem 7 when applied to the stochastic setting shows that it is possible to converge to the true mistake rate up to a constant factor, but only for the unique-label assumption. In this section, we show that in the stochastic setting with the general representation model, it is possible for the mistake rate to converge to the exception rate with no constant factors, and with a polynomial sample complexity. The proposed algorithm is not efficient; An open question for future work is whether this can be achieved with an efficient algorithm.

To prove the result, we propose a new general notion of *Feature Influence*, which may be of wider applicability. Consider some hypothesis class $\mathcal{H}_\Phi \subseteq \mathcal{Y}^\mathcal{X}$ defined over a set of features Φ, such that for any subset of the features $\Phi' \subseteq \Phi$, $\mathcal{H}_{\Phi'} \subseteq \mathcal{H}_\Phi$. Assume some feature feedback protocol. A *feature discovery protocol* is an algorithm that interacts with the environment and the teacher as specified in the protocol, and outputs some feature subset $\hat{\Phi} \subseteq \Phi$. Feature Influence, defined below, measures the effect on the distribution error of restricting the hypothesis class to features from $\hat{\Phi}$.

Formally, given a distribution $\mathcal{D}$, the error of a concept $h : \mathcal{X} \to \mathcal{Y}$ is $\text{err}(h, \mathcal{D}) := \mathbb{P}_{(X,Y)\sim\mathcal{D}}[h(X) \neq Y]$. For a sequence of labeled examples $S = ((x_1, y_1), \ldots, (x_n, y_n))$, the empirical error is $\text{err}(h, S) := \text{err}(h, \text{Unif}(S))$, where $\text{Unif}(S)$ is the uniform distribution over S. Given a class $\mathcal{H} \subseteq \mathcal{Y}^\mathcal{X}$, the smallest error that can be obtained by some $h \in \mathcal{H}$ on $\mathcal{D}$ is denoted $\text{err}(\mathcal{H}, \mathcal{D}) = \inf_{h \in \mathcal{H}} \text{err}(h, \mathcal{D})$.

Definition 1 (Feature Influence). *Fix an exception rate $\epsilon \in [0,1]$, an excess error target $\alpha \in [0,1]$, a confidence level $\delta \in (0,1)$, a mistake bound $b \in \mathbb{N}$, and a capacity bound $d \in \mathbb{N}$. Assume some feature feedback protocol with some class $\mathcal{H} := \mathcal{H}_\Phi$. The setting has α-Feature Influence with exception rate ϵ, confidence δ, mistake bound b and capacity bound d, if there exists a feature discovery protocol that outputs a feature subset $\hat{\Phi}$ such that for any stochastic i.i.d. stream which is consistent with the feature feedback protocol except for an exception rate of up to ϵ, if the protocol is applied to a prefix of an i.i.d. sample drawn according to the distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ induced by the setting and stops before making more than b mistakes, then:*

- *With probability at least $1 - \delta$, $\text{err}(\mathcal{H}_{\hat{\Phi}}, \mathcal{D}) \leq \epsilon + \alpha$,*

- $\mathbb{P}[\log_2 |\mathcal{H}_{\hat{\Phi}}| \leq d] = 1$.

The following theorem provides a sample complexity bound

for the stochastic setting for a general feature-based protocol and hypothesis class, using the notion of Feature Influence.

Theorem 9. *Suppose that a feature feedback setting with some $\mathcal{H}_\Phi$ has α-Feature Influence with exception rate ϵ, confidence δ, mistake bound b and capacity bound d. Then there exists an algorithm for the stochastic setting that obtains a mistake rate of at most $\epsilon + 2\alpha$ with probability at least $1 - 3\delta$ over i.i.d. streams of length at least*

$$\tilde{O}\left(\frac{b}{\alpha} + \left(\frac{\epsilon}{\alpha^3} + \frac{1}{\alpha^2}\right)(d + \log(1/\delta))\right).$$

The proof, provided in App. C, proves the above mistake bound for the following algorithm, where n is the length of the stream:

1. Run the feature discovery protocol that witnesses the Feature Influence with mistake limit b (thus observing some n_1 examples), and obtain the feature subset $\hat{\Phi}$. Set $\hat{\mathcal{H}} := \mathcal{H}_{\hat{\Phi}}$.

2. Observe the next $n_2 := \alpha n/2$ examples, predicting an arbitrary label for each example, and observing the true label provided by the teacher. Let S_2 be the sequence of n_2 labeled examples obtained this way. Find some $\hat{h} \in \operatorname{argmin}_{h \in \hat{\mathcal{H}}} \operatorname{err}(h, S_2)$.

3. Use $\hat{h}$ to predict labels for the rest of the input stream.

To apply this general result to our Discriminative Feature Feedback setting, we define an appropriate hypothesis class $\mathcal{H}_\Phi$ and prove a Feature Influence property. Let $\mathcal{H}_\Phi^{\mathrm{comp}} \subseteq \mathcal{Y}^{\mathcal{X}}$ be the set of all concepts that are consistent with some component representation of size at most m with features from Φ, as defined in Sec. 3. We have $\operatorname{err}(\mathcal{H}_\Phi^{\mathrm{comp}}, \mathcal{D}) \leq \epsilon$. We define an additional, more expressive, hypothesis class $\mathcal{H}_\Phi^+ \supseteq \mathcal{H}_\Phi^{\mathrm{comp}}$. $\mathcal{H}_\Phi^+$ is the class of concepts that can be described using a decision list with at most m rules, where each rule is a conjunction of up to $m - 1$ features with an associated label. Examples can be satisfied by more than one rule. The first rule in the decision list that is satisfied determines the label of the example. It is easy to see that all the possible true concepts c^* can be represented using a hypothesis from $\mathcal{H}_\Phi^+$. Hence, $\mathcal{H}_\Phi^+ \supseteq \mathcal{H}_\Phi^{\mathrm{comp}}$ and $\operatorname{err}(\mathcal{H}_\Phi^+, \mathcal{D}) \leq \operatorname{err}(\mathcal{H}_\Phi^{\mathrm{comp}}, \mathcal{D}) \leq \epsilon$.

To show a Feature Influence property, we first show that if $\mathcal{H}_\Phi^+$ is restricted to a subset of the features in an appropriate way, then the additional incurred error can be bounded. Let $\mathtt{Ex}$ be the event than an exception occurred in the current draw of $X \sim \mathcal{D}_X$. Denote $P[G] := \mathbb{P}[X \in G \wedge \neg\mathtt{Ex}]$. Recall that Φ is closed under negation. In every pair of features ϕ, ϕ' such that $\phi = \neg\phi'$, designate one of the two as the positive feature. Denote the positive feature that separates two components G and G' with different labels by $\phi_+(G, G')$. Note that $\phi_+(G, G') = \phi_+(G', G)$. Let P_ϕ be

the set of all sets of size 2 of components in $\mathcal{G}$ such that the two components are separated by ϕ in the representation:

$$P_\phi := \{\{G, G'\} \mid \phi_+(G, G') = \phi\}.$$

Denote

$$\beta_\phi := \sum_{\{G,G'\} \in P_\phi} P[G]P[G'],$$

and let $\Phi_\beta := \{\phi \mid \beta_\phi \geq \beta\}$. We first upper bound the effect of removing from Φ only features with a small β_ϕ.

Lemma 10. $\operatorname{err}(\mathcal{H}_{\Phi_\beta}^+, \mathcal{D}) \leq \epsilon + \sqrt{\beta} \cdot m^2/2$.

Proof. Consider the hypothesis $h_\beta \in \mathcal{H}_{\Phi_\beta}^+$, defined as a decision list with a rule for each component $G \in \mathcal{G}$, associated with the label $\ell(G)$. The rules are in descending order based on the marginal probability $P[G]$ for each $G \in \mathcal{G}$. The rule for G is a conjunction of all the discriminative features separating G from other components, except for features ϕ that have $\beta_\phi < \beta$. To bound the error of h_β, consider an example x in some component G. If x is predicted by h_β with a different label than $c^*(x)$, then it satisfies the rule of another component G' which has a larger (or equal) marginal probability than G and a different label. Moreover, the discriminative feature ϕ separating G and G' must satisfy $\beta_\phi < \beta$, or it would have been a part of the rule of G' and x would not have satisfied it. It follows that

$$\mathbb{P}[(X \in G \wedge \neg\mathtt{Ex}) \wedge (h_\beta(X) \neq c^*(X))]$$
$$\leq P[G] \cdot \mathbb{I}[\exists G' \in \mathcal{G}, \beta_{\phi_+(G,G')} \leq \beta \wedge P[G'] \geq P[G]]$$
$$\leq P[G] \cdot \sum_{G': \beta_{\phi_+(G,G')} \leq \beta} \mathbb{I}[P[G'] \geq P[G]].$$

Summing over all components, we have that

$$\mathbb{P}[\neg\mathtt{Ex} \wedge (h_\beta(X) \neq c^*(X))]$$
$$\leq \sum_{\{G,G'\} \subseteq \Phi: \beta_{\phi_+(G,G')} \leq \beta} \min\{P[G'], P[G]\}$$
$$= \sum_{\phi: \beta_\phi \leq \beta} \sum_{\{G,G'\} \in P_\phi} \min\{P[G'], P[G]\}.$$

For any given ϕ, we have

$$\sum_{\{G,G'\} \in P_\phi} \min\{P[G'], P[G]\} \leq \sum_{\{G,G'\} \in P_\phi} \sqrt{P[G']P[G]}$$
$$\leq \sqrt{|P_\phi| \sum_{\{G,G'\} \in P_\phi} P[G']P[G]}.$$

Let $\mathbf{1}$ be the all-1 vector of dimension $|P_\Phi|$, and let $\boldsymbol{v}$ be the vector of the same dimension with coordinate values $\sqrt{P[G']P[G]}$. By the Cauchy-Schwartz inequality,

$$\sum_{\{G,G'\} \in P_\phi} \sqrt{P[G']P[G]} = \langle \mathbf{1}, \boldsymbol{v} \rangle \leq \|\mathbf{1}\|_2 \|\boldsymbol{v}\|_2$$
$$= \sqrt{|P_\phi| \sum_{\{G,G'\} \in P_\phi} P[G']P[G]}.$$

Algorithm 3 Feature discovery protocol for Discriminative Feature Feedback

---

Input: Mistake limit b, threshold parameter $\beta \in (0,1)$
Output: A feature set $\hat{\Phi} \subseteq \Phi$
1: $t \leftarrow 0$; Get initial labeled example (x_0, y_0).
2: Initialize a function $F : \Phi \to \mathbb{N}$ to constant 0.
3: **while** $\|F\|_1 \equiv \sum_{\phi \in \Phi} F(\phi) \leq b/2$ **do**
4: $\quad$ $t \leftarrow t + 1$; Get next example x_t
5: $\quad$ Predict y_0 with explanation x_0.
6: $\quad$ Get true label y_t. $\quad \triangle$ *ignore additional feedback*
7: $\quad$ $x_{\text{base}} \leftarrow x_t, y_{\text{base}} \leftarrow y_t$.
8: $\quad$ **while** true **do**
9: $\quad\quad$ $t \leftarrow t + 1$; Get next example x_t
10: $\quad\quad$ Predict y_{base} with explanation x_{base}.
11: $\quad\quad$ **if** prediction was incorrect **then**
12: $\quad\quad\quad$ Get true label y_t and disc. feature ϕ_t.
13: $\quad\quad\quad$ $\phi_+ \leftarrow$ the positive feature out of $\phi_t, \neg \phi_t$.
14: $\quad\quad\quad$ $F(\phi_+) = F(\phi_+) + 1$.
15: $\quad\quad\quad$ **break from inner loop.**
16: Return $\hat{\Phi} := \{\phi \mid F(\phi) \geq \beta \|F\|_1\}$.

From the definition of β_ϕ, it follows that

$$\sum_{\{G,G'\} \in P_\phi} \min\{P[G'], P[G]\} \leq \sqrt{|P_\phi| \beta_\phi}.$$

Therefore,

$$\mathbb{P}[\neg \text{Ex} \wedge (h_\beta(X) \neq c^*(X))] \leq \sum_{\phi : \beta_\phi \leq \beta} \sqrt{|P_\phi| \beta_\phi}$$

$$\leq \sqrt{\beta} \sum_{\phi : \beta_\phi \leq \beta} \sqrt{|P_\phi|} \leq \sqrt{\beta} \cdot m^2/2.$$

The last inequality follows since $\sum_{\phi : \beta_\phi \leq \beta} \sqrt{|P_\phi|} \leq \sum_{\phi \in \Phi} |P_\phi| = \binom{m}{2}$. The proof is completed by noting that

$$\text{err}(\mathcal{H}^+_{\Phi_\beta}, \mathcal{D}) \leq \text{err}(h_\beta, \mathcal{D}) = \mathbb{P}[h_\beta(X) \neq c^*(X)]$$

$$\leq \mathbb{P}[\text{Ex}] + \mathbb{P}[\neg \text{Ex} \wedge (h_\beta(X) \neq c^*(X))],$$

and $\mathbb{P}[\text{Ex}] \leq \epsilon$. $\quad\square$

Alg. 3 provides a feature discovery protocol that identifies features with a large β_ϕ. It repeatedly sets an observed labeled example as a baseline, and uses it to predict the label of the next examples, until a prediction is incorrect and provides feature feedback. The output $\hat{\Phi}$ includes all the features observed at least a β fraction of the time. We now show that with a high probability, this protocol identifies all features with a large β_ϕ. Note that only a single feature feedback is obtained for each instance of $(x_{\text{base}}, y_{\text{base}})$. The distribution of the examples x_{base} used for prediction is equal to $\mathcal{D}_X$. The distribution of examples x_t that get an incorrect answer, given x_{base}, is equal to $\mathcal{D}_X$ conditioned

on $y_t \neq y_{\text{base}}$. Thus, for any $\phi \in \Phi$ and a feature feedback ϕ_t, we have

$$\mathbb{P}[\phi_t = \phi]$$

$$= \sum_{G \in \mathcal{G}} P[x_{\text{base}} \in G] \sum_{G' : \{G, G'\} \in P_\phi} \mathbb{P}[x_t \in G' \mid y_t \neq y_{\text{base}}]$$

$$\geq \sum_{G \in \mathcal{G}} P[G] \sum_{G' : \{G, G'\} \in P_\phi} P[G'] \geq 2\beta_\phi. \quad (1)$$

Moreover, the collected ϕ_t are statistically independent. The following lemma follows (see proof in App. D).

Lemma 11. *Suppose that $\|F\|_1 \geq 6 \log(1/(2\delta\beta))/\beta$. Then with a probability at least $1 - \delta$, $\Phi_\beta \subseteq \hat{\Phi}$.*

We can now conclude a Feature Influence property for our Discriminative Feature Feedback component model.

Theorem 12. *Let $\mathcal{H} = \mathcal{H}^+_\Phi$ be the hypothesis class defined above. For any $\beta, \delta, \epsilon \in (0,1)$, the Discriminative Feature Feedback setting with a representation size m has $\sqrt{\beta} \cdot m^2/2$-Feature Influence with exception rate ϵ, confidence δ, mistake bound $12 \log(1/(2\delta\beta))/\beta$ and capacity bound $m \log(m) + m^2 \log(3/\beta)$.*

Proof. The feature discovery protocol in Alg. 3 makes at most two mistake for every ϕ_t it obtains. Thus, it obtains at least $b/2$ independent random samples of features ϕ. Setting $b = 12 \log(1/(2\delta\beta))/\beta$, it follows from Lemma 11 that with probability $1 - \delta$ we have $\Phi_\beta \subseteq \hat{\Phi}$, so $\mathcal{H}^+_{\Phi_\beta} \subseteq \mathcal{H}^+_{\hat{\Phi}}$. By Lemma 10, $\text{err}(\mathcal{H}^+_{\hat{\Phi}}, \mathcal{D}) \leq \text{err}(\mathcal{H}^+_{\Phi_\beta}, \mathcal{D}) \leq \epsilon + \sqrt{\beta} \cdot m^2/2$, as required by the definition of Feature Influence.

To prove the capacity bound, note that $|\hat{\Phi}| \leq 1/\beta$. Therefore, the size of $\mathcal{H}_{\hat{\Phi}}$ can be bounded by the total number of possible decision lists with up to m rules, up to $m - 1$ features in each rule, and up to m different labels, to yield $|\mathcal{H}_{\hat{\Phi}}| \leq (2/\beta + 1)^{m(m-1)} \cdot m^m$. Thus, $\log_2 |\mathcal{H}_{\hat{\Phi}}| \leq m \log(m) + m^2 \log(3/\beta)$. $\quad\square$

Combining this result with the general Feature Influence result of Theorem 9, we get the following guarantee.

Corollary 13. *In the stochastic setting, there exists an algorithm that obtains a mistake rate of at most $\epsilon + \alpha$ with probability at least $1 - \delta$ for i.i.d. streams of length at least*

$$\tilde{O}\left(\frac{m^4 + \log(1/\delta)}{\alpha^3}\right).$$

Proof. The claim immediately follows by setting $\beta := \alpha^2/m^4$ in Theorem 12, which gives an $\alpha/2$-Feature Influence with exception rate ϵ, confidence δ, mistake bound $\tilde{O}((m^4 + \log(1/\delta)/\alpha^2)$ and capacity bound $\tilde{O}(m^2 \log(1/\alpha))$, and applying Theorem 9. $\quad\square$

Thus, using the algorithm proposed above, the mistake rate converges to the exception rate for $n \to \infty$, as desired.

6 CONCLUSIONS

We showed that robust interactive learning with Discriminative Feature Feedback can be achieved with significantly improved mistake bound compared to previous results, both in the adversarial setting and in the stochastic setting. Several interesting open problems remain for future work, such as the existence of an efficient algorithm for the stochastic setting that converges to the exception rate, and the possibility of improving the mistake bound in the general adversarial setting to match the adversarial unique-label setting.

Acknowledgements

References

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, 2010.

W. Croft and R. Das. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 349–368, 1990.

A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the ERM principle. *Journal of Machine Learning Research*, 16:2377–2404, 2015.

S. Dasgupta and S. Sabato. Robust learning from discriminative feature feedback. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 973–982. PMLR, 2020.

S. Dasgupta, A. Dey, N. Roberts, and S. Sabato. Learning from discriminative feature feedback. In *Advances in Neural Information Processing Systems*, pages 3955–3963, 2018.

G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of ACM Special Interest Group on Information Retrieval*, 2008.

L. Guo, E. M. Daly, O. Alkan, M. Mattetti, O. Cornec, and B. Knijnenburg. Building trust in interactive machine learning via user contributed interpretable rules. In *27th International Conference on Intelligent User Interfaces*, IUI '22, pages 537–548, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391443.

S. Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

W. Liang, J. Zou, and Z. Yu. Alice: Active learning with contrastive natural language explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4380–4391, 2020.

O. Mac Aodha, S. Su, Y. Chen, P. Perona, and Y. Yue. Teaching categories to human learners with visual explanations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

S. Poulis and S. Dasgupta. Learning with feature feedback. In *Twentieth International Conference on Artificial Intelligence and Statistics*, 2017.

H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 841–846, 2005.

P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

B. Settles. Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In *Empirical Methods in Natural Language Processing*, 2011.

S. Teso and K. Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 239–245, 2019.

R. Visotsky, Y. Atzmon, and G. Chechik. Learning with per-sample side information. In *AGI*, 2019.

J. Zou, K. Chaudhuri, and A. T. Kalai. Crowdsourcing feature discovery via adaptively chosen comparisons. In *Conference on Human Computation and Crowdsourcing (HCOMP)*, 2015.

Algorithm 4 `PFR-DFF`: A parameter-free robust Discriminative Feature Feedback algorithm for the adversarial setting

---
1: $v \leftarrow 1$
2: **while** true **do**
3:     $\widetilde{m} \leftarrow 1$.
4:     **while** $\text{UB}(\widetilde{m}, 0) \leq v$ **do**
5:         $\widetilde{k} \leftarrow 0$
6:         **while** $\text{UB}(\widetilde{m}, \widetilde{k}) \leq v$ **do**
7:             **if** $\text{UB}(\widetilde{m}, \widetilde{k}) > v/2$ **then**
8:                 Run `SR-DFF`$(\widetilde{m}, \widetilde{k})$ on the next examples in the stream;
9:                 `Break` from the loop if more than $\text{UB}(\widetilde{m}, \widetilde{k})$ mistakes have occured.
10:         $\widetilde{k} \leftarrow 2\widetilde{k} + 1$
11:     $\widetilde{m} \leftarrow 2\widetilde{m}$
12:     $v \leftarrow 2v$

A A Parameter-free algorithm (Proof of Theorem 2)

The algorithm `SR-DFF`, presented in Sec. 4.1, requires a number of components m as input. Its mistake bound depends on m and on k, the minimal number of exceptions for a representation with at most m components. We now show that a nested doubling trick allows applying `SR-DFF` even if no upper bound m is known. The cost is only a logarithmic factor in the number of mistakes. A similar approach can be applied for the stochastic algorithm described in Sec. 5.

The parameter-free algorithm `PFR-DFF` is provided in Alg. 4. We now prove Theorem 2 by bounding the number of mistakes made by `PFR-DFF` for all pairs of m and k such that the input stream is consistent with some representation of size m with at most k exceptions.

Proof of Theorem 2. The upper bounds of m and k in Theorem 1 hold also for any sub-sequence of the input sequence. Therefore, once `PFR-DFF` runs `SR-DFF`(m', k') for some $m' \geq m, k' \geq k$, it will never break from the loop. This will occur for some $m' < 2m$, $k' \leq 2k$, when $v = v'$ for some $v' \leq 2\text{UB}(m', k') \leq 2\text{UB}(2m, 2k) \leq 8\text{UB}(m, k)$, where the last inequality follows from the definition of UB.

In a loop over $\widetilde{k}$ with given v and $\widetilde{m}$, the number of rounds is $J \leq \text{argmax}\{j \geq 0 \mid \text{UB}(\widetilde{m}, 2^j - 1) \leq v\}$. The total number of mistakes made by `PFR-DFF` in this loop is upper-bounded by

$$\sum_{j=1}^{J}(1 + \text{UB}(\widetilde{m}, 2^j - 1)) = (J+1)(1 + \widetilde{m}(\widetilde{m} - 1)) + \widetilde{m}\sum_{j=1}^{J}(2^j - 1) \leq (J+1)(1 + \widetilde{m}(\widetilde{m} - 1)) + \widetilde{m}(2^{J+1} - 2).$$

Let $\hat{k} = 2^J - 1$. By the definition of J, $\text{UB}(\widetilde{m}, \hat{k}) < v$. Thus, the RHS of the inequality above is upper-bounded by

$$(\log_2(\hat{k} + 1) + 1)(1 + \widetilde{m}(\widetilde{m} - 1)) + 2\widetilde{m}\hat{k} \leq (\log_2(\hat{k}) + 2) \cdot (\text{UB}(\widetilde{m}, \hat{k}) + 1) \leq (\log_2(\hat{k}) + 2) \cdot (v + 1),$$

where the penultimate inequality follows from the definition of UB. To upper bound $\hat{k}$, note that since $\text{UB}(\widetilde{m}, \hat{k}) \leq v$, by the definition of UB, $\hat{k} \leq v$. Therefore, for given $v, \widetilde{m}$, the total number of mistakes is at most $(\log_2(v) + 2) \cdot (v + 1)$. From the definition of UB, we have that for a given value of v, the maximal value of $\widetilde{m}$ that `PFR-DFF` might try satisfies $\widetilde{m} \leq v$, since $\text{UB}(\widetilde{m}, 0) = \widetilde{m}(\widetilde{m} - 1) \leq v$. Therefore, the number of rounds over $\widetilde{m}$ for a given v is at most $\log_2(v) + 1$. It follows that the total number of mistakes for loop with a given v is at most $(\log_2(v) + 1)(\log_2(v) + 2) \cdot (v + 1)$.

Summing over all values of v that `PFR-DFF` tries until reaching v', we get a total mistake bound of

$$\sum_{i=0}^{\log_2(v')} (i+1)(i+2)(2^i + 1) \leq 4v'\log^2(v') \leq 32\text{UB}(m, k)\log^2(8\text{UB}(m, k)),$$

as claimed. $\qquad\square$

B Proof of Theorem 7

Consider a specific run of the algorithm. To upper bound the total number of mistakes, we distinguish the types of mistakes:

- A mistake on an exception x_t; There are at most k such mistakes.

- A mistake on a valid example:

  - A mistake that caused an update (of any kind) in a rule created by an exception (an exception rule). Let M_{ev} denote the number of mistakes in exception rules by valid examples.
  - A mistake that caused the refinement of a rule by adding a feature. Let A_{vv} denote the number of additions of features to valid rules by valid examples.
  - A mistake that caused an attempt to create a rule (that is, a random draw of a bit B). Denote the number of attempts to create a rule by a valid example by $\tilde{R}_v$.
  - A mistake that caused a deletion of a feature from a valid rule. Let D_{vv} denote the number of deletions of features from valid rules by valid examples.

The total number of mistakes M_T made by the algorithm is thus upper bounded by:

$$M_T \leq k + M_{ev} + A_{vv} + \tilde{R}_v + D_{vv}.$$

Let R_v be the number of rules that were actually created based on a valid examples (cases where $B = 1$). We have $\mathbb{E}[R_v] = p\mathbb{E}[\tilde{R}_v]$, thus, rearranging, we have

$$\mathbb{E}[M_T] \leq k + \mathbb{E}[R_v]/p + \mathbb{E}[A_{vv}] + \mathbb{E}[D_{vv}] + \mathbb{E}[M_{ev}]. \tag{2}$$

Next, we bound A_{vv} and D_{vv}. Each valid rule has at most $m - 1$ true discriminative features. However, true features might be deleted from a valid rule by an exception and could then be re-added. Denote by D_{ve} the total number of true discriminative features deleted from valid rules by exceptions. It follows that $A_{vv} \leq (m-1)R_v + D_{ve}$. To bound D_{vv}, note that every incorrect feature in a valid rule deleted by a valid example must have been added by an exception. Denote by A_{ve} the total number of incorrect features added to some valid rule by an exception. Then $D_{vv} \leq A_{ve}$. It follows that

$$\mathbb{E}[M_T] \leq k + \mathbb{E}[R_v](1/p + (m-1)) + \mathbb{E}[D_{ve}] + \mathbb{E}[A_{ve}] + \mathbb{E}[M_{ev}]. \tag{3}$$

Next, we upper bound R_v. The algorithm only creates a rule with label y_t if such a rule does not currently exist. There are at most m valid labels, thus the total number of valid rules created during the run is at most m plus the number of deleted valid rules. To bound the number of deleted valid rules, observe that a rule is only deleted if at least $m + l - 1$ updates, as counted by the counter U, have been applied to it. Lemma 8, stated in Sec. 4.3, links the number of rule updates to the number of exceptions that caused updates to the rule. Since the number of exceptions is bounded, this will lead to a bound on the number of deleted rules. We now give the proof of the lemma.

Proof of Lemma 8. Let a be the total number of features that were added to $C[x]$ until the end of iteration t, and let b be the number of features that were deleted. Let c be the number of examples that were ignored due to failing the condition on line 9. Note that $U(x) = a + b + c$. All of the ignored examples must have been exceptions, thus at least c updates were caused by exceptions.

For every deletion of a feature from a valid rule, at least one of its addition and its deletion must have been caused by an exception, since they cannot both be correct. Therefore, the number of exceptions that have led to additions or deletions in this rule so far is at least b. In addition, out of the a features that were added to the rule, at most $m - 1$ can be correct. Therefore, at least $a - (m - 1)$ of the existing features were caused by exceptions.

It follows that the total number of exceptions that caused updates to this rule is at least

$$c + \max(a - (m-1), b) \geq (a + b + c - (m-1))/2 \geq (U(x) - (m-1))/2.$$

This completes the proof. □

To bound R_v, note that when a rule $C[x]$ is deleted, it satisfies $U(x) = m - 1 + l$. Therefore, if the rule is valid, it follows from Lemma 8 that it has been updated by at least $l/2$ exceptions. Let U_{ve} be the number of updates to valid rules by exceptions. Then the total number of deletions of valid rules is at most $U_{ve}/(l/2)$, hence $R_v \leq m + 2U_{ve}/l$. In addition, $D_{ve} + A_{ve} \leq U_{ve}$. Thus, from Eq. (3),

$$\mathbb{E}[M_T] \leq k + ((m + 2U_{ve}/l)(1/p + m - 1)) + \mathbb{E}[U_{ve}] + \mathbb{E}[M_{ev}]$$
$$= k + m/p + m^2 - m + (2(1/p + m - 1)/l + 1)\mathbb{E}[U_{ve}] + \mathbb{E}[M_{ev}]. \tag{4}$$

Next, we upper bound M_{ev}. Let R_e be the number of created exception rules. An exception rule is deleted if it accumulates $m + l - 1$ updates, hence $M_{ev} \leq R_e(m + l - 1)$. In addition, letting $\tilde{R}_e$ be the number of attempts to create an exception rules (draws of B), we have $U_{ve} + \tilde{R}_e \leq k$. In addition, $\mathbb{E}[R_e] = p\mathbb{E}[\tilde{R}_e]$. Therefore,

$$\mathbb{E}[U_{ve}] \leq k - \mathbb{E}[\tilde{R}_e] = k - \mathbb{E}[R_e]/p.$$

Combining these bounds with Eq. (4), we get

$$\mathbb{E}[M_T] \leq k + m/p + m^2 - m + (2(1/p + m - 1)/l + 1)(k - \mathbb{E}[R_e]/p) + \mathbb{E}[R_e](m + l - 1)$$
$$= m/p + m^2 - m + (2(1/p + m - 1)/l + 2)k$$
$$+ (m + l - 1 - \frac{1}{p}(2(1/p + m - 1)/l + 1))\mathbb{E}[R_e].$$

Setting $l = 1/p$, the coefficient of $\mathbb{E}[R_e]$ becomes

$$m + 1/p - 1 - 2(m + 1/p - 1) - 1/p = -m - 2/p + 1 < 0,$$

and the coefficient of k is $2(1/p + m - 1)/l + 2 = 4 + 2p(m - 1)$.

Setting $p = 1/(m - 1)$, it follows that $\mathbb{E}[M_T] \leq 2m(m - 1) + 6k$. This completes the proof of Theorem 7.

C Proof of Theorem 9

Consider the 3-stage algorithm described after the statement of Theorem 9. Let $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ be the distribution of labeled examples from which the i.i.d. stream is drawn. The expected mistake rate during Stage 3 of the algorithm is $\mathrm{err}(\hat{h}, \mathcal{D})$. The length of Stage 3 is $n_3 = n - n_1 - n_2$. By Hoeffding's inequality, with probability at least $1 - \delta$, the total number of mistakes of $\hat{h}$ during Stage 3 is at most $n_3 \cdot \mathrm{err}(\hat{h}, \mathcal{D}) + \sqrt{n_3 \log(1/\delta)/2}$. Since $n_3/n < 1$, the mistake rate over all the stages is at most $(b + n_2)/n + \mathrm{err}(\hat{h}, \mathcal{D}) + \sqrt{\log(1/\delta)/(2n)}$.

Since $\hat{h}$ is the empirical risk minimizer of $\hat{\mathcal{H}}$ over the i.i.d. sample $S \sim \mathcal{D}^{n_2}$, we have, for the case of binary classification ($|\mathcal{Y}| = 2$) (Boucheron et al., 2005), that with a probability of at least $1 - \delta$ over the randomness of the input sample during Stage 2,

$$\mathrm{err}(\hat{h}, \mathcal{D}) \leq \mathrm{err}(\hat{\mathcal{H}}, \mathcal{D}) + \tilde{O}\left(\sqrt{\mathrm{err}(\hat{\mathcal{H}}, \mathcal{D})\frac{d + \log(1/\delta)}{n_2}} + \frac{d + \log(1/\delta)}{n_2}\right).$$

A standard reduction argument from multiclass learning to binary classification (Daniely et al., 2015) shows that the inequality above holds also for a general finite $\mathcal{Y}$.

From the definition of Feature Influence, we have that with a probability at least $1 - \delta$, $\mathrm{err}(\hat{\mathcal{H}}, \mathcal{D}) \leq \epsilon + \alpha$. Therefore, with a probability at least $1 - 3\delta$, the total mistake rate of the algorithm over all the stages is at most

$$\frac{b + n_2}{n} + \epsilon + \alpha + \tilde{O}\left(\sqrt{\frac{(\epsilon + \alpha)(d + \log(1/\delta))}{n_2}} + \frac{d + \log(1/\delta)}{n_2}\right) + \sqrt{\frac{\log(1/\delta)}{2n}} =$$
$$\epsilon + \frac{3}{2}\alpha + \frac{b}{n} + \tilde{O}\left(\sqrt{\left(\frac{\epsilon}{\alpha} + 1\right)\frac{(d + \log(1/\delta))}{n}} + \frac{d + \log(1/\delta)}{\alpha n}\right).$$

To complete the proof, note that a stream length of

$$n = \tilde{O}\left(\frac{b}{\alpha} + \left(\frac{\epsilon}{\alpha^3} + \frac{1}{\alpha^2}\right)(d + \log(1/\delta))\right)$$

suffices for the expression above to be $\epsilon + 2\alpha$.

D Proof of Lemma 11

Fix some $\phi \in \Phi$, and let $\hat{p}_\phi := F(\phi)/\|F\|_1$. By Eq. (1), $\mathbb{E}[\hat{p}_\phi] \geq 2\beta_\phi$. Hence, by Bernstein's inequality, with a probability at least $1 - 2\delta\beta$,

$$\hat{p}_\phi \geq 2\beta_\phi - \frac{2\log(1/(2\delta\beta))}{3\|F\|_1} - \sqrt{\frac{2\beta\log(1/(2\delta\beta))}{\|F\|_1}}.$$

Since $\|F\|_1 \geq 6\log(1/(2\delta\beta))/\beta$ and $\beta_\phi \geq \beta$, this implies that $\hat{p}_\phi \geq 2\beta_\phi - \beta \geq \beta$.

Now, note that

$$\sum_{\phi \in \Phi} \beta_\phi = \sum_{\{G,G'\} \subseteq \mathcal{G}} \mathbb{P}[G]\mathbb{P}[G'] = \frac{1}{2}\sum_{G \in \mathcal{G}} \mathbb{P}[G] \cdot \sum_{G' \in \mathcal{G}\setminus\{G\}} \mathbb{P}[G'] \leq \frac{1}{2}.$$

Therefore, there are at most $1/(2\beta)$ features ϕ such that $\beta_\phi \geq \beta$. By a union bound, this holds simultaneously for all ϕ such that $\beta_\phi \geq \beta$, with probability $1 - \delta$.