Runaway Feedback Loops in Predictive Policing*

Danielle Ensign DANIPHYE@GMAIL.COM
University of Utah

Sorelle A. Friedler SORELLE@CS.HAVERFORD.EDU
Haverford College

Scott Neville DROP.SCOTT.N@GMAIL.COM
University of Utah

Carlos Scheidegger CSCHEID@CSCHEID.NET
University of Arizona

Suresh Venkatasubramanian† SURESH@CS.UTAH.EDU
University of Utah

Editors: Sorelle A. Friedler and Christo Wilson

Abstract

Predictive policing systems are increasingly used to determine how to allocate police across a city in order to best prevent crime. Discovered crime data (e.g., arrest counts) are used to help update the model, and the process is repeated. Such systems have been empirically shown to be susceptible to runaway feedback loops, where police are repeatedly sent back to the same neighborhoods regardless of the true crime rate.

In response, we develop a mathematical model of predictive policing that proves why this feedback loop occurs, show empirically that this model exhibits such problems, and demonstrate how to change the inputs to a predictive policing system (in a black-box manner) so the runaway feedback loop does not occur, allowing the true crime rate to be learned. Our results are quantitative: we can establish a link (in our model) between the degree to which runaway feedback causes problems and the disparity in crime rates between areas. Moreover, we can also demonstrate the way in which *reported* incidents of crime (those reported by residents) and *discovered* incidents of crime (i.e. those directly observed by police officers dispatched as a result of the predictive policing algorithm) interact: in brief, while reported incidents can attenuate the degree of runaway feedback, they cannot entirely remove it without the interventions we suggest.

Keywords: Feedback loops, predictive policing, online learning.

1. Introduction

Machine learning models are increasingly being used to make real-world decisions, such as who to hire, who should receive a loan, where to send police, and who should receive parole. These deployed models mostly use traditional batch-mode machine learning, where decisions are made and observed results supplement the training data for the next batch.

However, the problem of *feedback* makes traditional batch learning frameworks both inappropriate and (as we shall see) incorrect. Hiring algorithms only receive feedback on people who were hired, predictive policing algorithms only observe crime in neighborhoods they patrol, and so on. Decisions made by the system influence the data that is fed to it in the future. For example, once a decision has been made to patrol a certain neighborhood, crime discovered in *that* neighborhood will be fed into the training apparatus for the next round of decision-making.

In this paper, we focus on predictive policing – an important exemplar problem demonstrating

* This research was funded in part by the NSF under grants IIS-1633387, IIS-1513651, and IIS-1633724. Code for our urn simulations can be found at https://github.com/algofairness/runaway-feedback-loops-src.

† Corresponding author.

these feedback issues. Predictive policing is increasingly employed to determine where to send police, who to target for surveillance, and even who may be a future crime victim (Perry, 2013). We focus on the most popular of these forms of predictive policing (with `PredPol`, HunchLab, IBM, and other companies entering the market) which attempts to determine how to deploy police given historical crime data.

Definition 1 (Predictive Policing) *Given historical crime incident data for a collection of regions, decide how to allocate patrol officers to areas to detect crime.*

Once police are deployed based on these predictions, data from observations in the neighborhood is then used to further update the model. We will call these observations *discovered incidents*, as opposed to *reported incidents* that are crime incidents reported to the police (e.g., via 911 calls). Since such discovered incidents only occur in neighborhoods that police have been sent to *by the predictive policing algorithm itself*, there is the potential for this sampling bias to be compounded, causing a runaway feedback loop. Indeed, Lum and Isaac (2016) have shown that this can happen.

Lum and Isaac's work focused on `PredPol` (Mohler et al., 2015), a predictive policing system in use by the LAPD and other cities across the U.S.. Lum and Isaac (2016) model what would happen if `PredPol` were used in Oakland to distribute police to find drug crime by using historical crime incident data as the historical data and a synthetic population of likely drug users based on public health data U.S. DoJ via ICPSR (2015); U.S. HHS via ICPSR (2015); they find that increasing policing efforts based on discovered incidents causes `PredPol`'s prediction to substantially diverge from the true crime rate, repeatedly sending back police to the same neighborhoods.

In addition to its importance in the criminal justice pipeline, predictive policing serves as an archetypal problem, through which we can better understand issues which arise out of deploying batch-mode machine learning algorithms in an online setting, where they essentially see results that are influenced by their own predictions. Other such algorithms include recidivism prediction, hiring algorithms, college admissions, and distribution of loans. In all of these contexts, the outcome of the prediction (e.g., who to hire) determines what feedback the algorithm receives (e.g., who performs well on the job).

1.1. Results

We use the theory of urns (a common framework in reinforcement learning) to analyze existing methods for predictive policing. We show formally as well as empirically why these methods will not work. Subsequently, we provide remedies that can be used directly with these methods in a black-box fashion that improve their behavior, and provide theoretical justification for these remedies.

2. Related Work

Our work builds most strongly on the work of Lum and Isaac (2016) described above, demonstrating the consequences of feedback loops in simulation in the predictive policing setting. There are a number of systems currently in place for predictive policing Perry (2013). The most well known system used for predictive policing is called `PredPol` (Mohler et al., 2015) (described in more depth below). Our implementation of `PredPol` is the one used by Lum and Isaac (2016) in their work. Recidivism prediction systems are also related to this work in that we believe they may exhibit some of the same feedback loop issues, given that recidivism outcomes are only known for prisoners who are released. While the details of the actual implementations (such as COMPAS NorthPointe (2012)) remain proprietary, Berk and Bleich (2013) provide a comprehensive review of the methods used in this area.

2.1. `PredPol`

The predictive policing software `PredPol` will be critical to our experimental investigations, so we describe it in more detail here. `PredPol` (Mohler et al., 2015) assumes that crimes follow an earthquake aftershock model, so that regions that previously experienced crime are likely to experience crime again, with some decay. Mohler et al. (2015) model the crime rate $\lambda_r(t)$ in region r at time t as follows: $\lambda_r(t) = \mu_r + \sum_{t_r^i < t} \theta\omega e^{-\omega(t-t_r^i)}$

where t_n^i represents the time of an event in region r, ω quantifies the time decay of a shock, and θ captures the degree to which aftershocks are generated from an initial event. They use an expectation-maximization procedure to determine the parameters of the model.

Note that this model only uses incident data (including both discovered and reported incidents – see Section 3.1) per region to determine the true crime rate[1] and does not use any context in the form of demographics, arrest profiles and so on. `PredPol`, in essence, is predicting where incidents will be reported or discovered (since that's all it sees), not where *crime* will happen. Each day officers are sent to the areas with highest predicted intensity and the resulting discovered incident data is fed back into the system.

3. Predictive Policing with Urns

We will model the predictive policing process by a series of urn models with increasing complexity. Urn models (especially the Pólya-Eggenberger urns) have a long history in machine learning, but notably also in reinforcement learning (Pemantle, 2007), where they have been used, starting with the work of Erev and Roth (1998), as a way to model how bounded-rationality players in a game might interact with each other. Studying the dynamics of urn models allows us to understand the convergent behavior of reinforcement learning in such settings.

We will use a *generalized* Pólya urn model (Pemantle, 2007) containing balls of two colors (red and black). At each time step, one ball is drawn from the urn, the color is noted, and the ball is replaced. Then the following replacement matrix is used to decide how to update the urn contents:

	Red addition	Black addition
Sample red	a	b
Sample black	c	d

This matrix says that if we initially sampled a red ball, then we replace it and add a more red balls and b more black balls to the urn. We refer to the *standard Pólya urn* as a generalized urn with $a = d = 1$ and $b = c = 0$.

1. `PredPol`— critically — conflates amount of crime and incident data.

3.1. Goals and assumptions

In the simplest predictive policing setting, a precinct has a single police officer and polices two regions A and B. Every day the police officer is sent to one neighborhood where they may or may not observe an incident; if they do, it is logged and we refer to such a log as a *discovered* incident. In addition, residents might report incidents that are also logged: we call these *reported* incidents. The goal is to build a predictive model for where to send the officer on each day. Specifically, the goal is to distribute the police officers in proportion to the crime in each area.[2]

Goal 3.1 (Effective Policing) *A region with Λ percent of the crime in the precinct should receive Λ percent of the police.*

Achieving this goal requires learning the relative crime rates of the regions.

To understand the behavior of predictive models, we will make some simplifying assumptions. We will firstly assume that the predictive model only uses current statistics (in some form) to make predictions.

Assumption 3.1 (Predictive Model) The officer tosses a coin based on current statistics to decide where to go next.

To fully specify a predictive model, we also need to understand *context* – what information is collected during policing – and *ground truth* – what assumptions we make about underlying crime rates. We assume the simplest form of context.

Assumption 3.2 (Context) The only information retained about a crime is a count.

Assumptions about ground truth are both critical and complicated. For some neighborhood A, let λ_A be the underlying ground truth crime rate for the neighborhood. We will assume that this is observed via discovered and reported incidents. Let d_A be the rate at which police that visit neighborhood A discover incidents. Let w_d be the weight of the discovered incidents within all

2. Why should this be the goal? Suppose there are exactly enough police officers to stop all the crime and no more, then a deployment according to the true crime rates will perfectly police all regions.

incidents. Similarly, let r_A be the rate at which incidents are reported from neighborhood A, and let w_r be the weight of reported incidents among all incidents. We will assume that $w_r + w_d = 1$. The total rate of incident data from neighborhood A is then $w_d \cdot d_A + w_r \cdot r_A$. We note here that *discovered* incidents are directly implicated in the feedback loop since police are deployed in areas based on the results of the predictive model. *Reported* incidents on the other hand are not.

To start our examinations, we make the following assumptions. In the subsections below, we explore what happens as we vary these factors.

Assumption 3.3 (Truth in Crime Data) If an officer goes to a location A with an underlying ground truth crime rate of λ_A, the officer discovers crime at a rate of λ_A. I.e., $d_A = \lambda_A$. Reported incidents are also reported at a rate that matches the underlying ground truth crime rate, i.e., $r_A = \lambda_A$.

Note that Assumption 3.3 allows the predictive policing system to operate in a generous context. There are many reasons to believe that this assumption does not hold. We will show that even in this optimistic setting problems occur.

Assumption 3.4 (Discovery Only) Incident data is only collected by an officer's presence in a neighborhood. Neighborhoods with no officers will contribute no incidents to the data. I.e., $w_d = 1$ and $w_r = 0$.

We will also start with the assumption that all incident data is made up of discovered incidents. We will modify this assumption to also account for reported incidents in Section 3.4.

3.2. Uniform crime rates

Let us start by assuming that the crime rate is uniform between areas.

Assumption 3.5 (Uniform Crime Rate) If an officer goes to a location, crime happens with probability λ. I.e., for any neighborhoods A and B, $\lambda_A = \lambda_B = \lambda$.

Consider an urn that contains red and black balls, where the proportion of red and black balls represent the current observed statistics of crime in areas A and B respectively. Visiting area A corresponds to picking a red ball and visiting area B corresponds to picking a black ball. Observing crime (which happens with probability λ) causes a new ball of the same color to be placed in the urn. The initial balls are always returned to the urn. The long-term distribution of red and black balls in the urn corresponds to the long-term belief about crime prevalence in the two areas.

In general, we can describe the evolution of this process as the following urn. We toss a coin that returns 1 with probability λ. If the coin returns 1, we simulate one time step of a standard Pólya urn, and if 0, we merely replace the sampled ball. This corresponds to a standard Pólya urn "slowed" down by a factor λ. As such, its long-term convergence is well-characterized. Let the beta distribution $\text{Beta}(\alpha, \beta)$ be a distribution over the interval $[0, 1]$ where the probability of x is given by[3] $f(x; \alpha, \beta) \propto x^{\alpha-1}(1-x)^{\beta-1}$

Lemma 2 (Renlund (2010)) *Assume the urn starts off with n_r red balls and n_b black balls. Then the limiting probability of seeing a red ball is a draw from the beta distribution $\text{Beta}(n_r, n_b)$.*

Significance. The long-term probability of seeing red is the long-term estimate of crime in area A *generated by the model*. The above result shows that this probability is a random draw governed only by the parameters n_r, n_b, which represents the prior *belief* of the system. In other words, the prior belief coupled with the lack of feedback about the unobserved region *prevents the system from learning that the two areas are in fact identical with respect to crime rates.*

On the contrary, consider how this process would work *without* feedback. The officer could be sent to an area chosen uniformly at random each day, and this process would clearly converge to a uniform crime rate for each area. Indeed, such a process resembles the standard update for the bias of a coin where the prior distribution on the bias is governed by a Beta distribution.

3.3. Non-uniform crime rates

Let us now drop the assumption of uniformity in crime rates, replacing Assumption 3.5 by

3. The constant of proportionality is $\Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ where $\Gamma(x)$ is the standard gamma function.

Assumption 3.6 (Non-uniform Crime Rate) A visit to area A has probability λ_A of encountering a crime, and a visit to area B has probability λ_B of encountering a crime.

Nonuniform crime rates in neighborhoods A and B can also be modeled by a Pólya urn, with the caveat that the updates to the urn are now random variables instead of deterministic updates. Formally, we can think of the urn as being described by the 2×2 (addition) matrix

$$\begin{pmatrix} X_A & 0 \\ 0 & X_B \end{pmatrix}$$

where X_A is a Bernoulli variable taking the value 1 with probability λ_A and 0 with probability $1 - \lambda_A$, and X_B is defined similarly

If the urn satisfied the so-called *balance* condition that the number of balls added at each time step is a constant (Mahmoud and Morcrette, 2012), then we could invoke standard results to determine the limiting behavior. This is not the case in this setting. However, we now show that it is possible to reduce this to a deterministic update model by exploiting the Bernoulli structure of the update.

At any time t, let $n_A^{(t)}, n_B^{(t)}$ be the number of balls "colored" A and B respectively. The probability of adding *any* ball to the urn is given by the expression

$$\frac{n_A^{(t)}\lambda_A + n_B^{(t)}\lambda_B}{n_A^{(t)} + n_B^{(t)}}$$

Note that this can be viewed as a convex combination of the two probabilities λ_A and λ_B and so the overall probability of a ball being added to the bin varies between two constants.

As before, consider the update process limited to time steps when a ball is added to the urn. The probability of adding a ball colored A, *conditioned on adding some ball*, is given by

$$\frac{\Pr(\text{adding a A-colored ball})}{\Pr(\text{adding some ball})} = \frac{n_A^{(t)}\lambda_A}{n_A^{(t)}\lambda_A + n_B^{(t)}\lambda_B}$$

with a similar expression for adding a B-colored ball.

This is identical to the deterministic Pólya urn in which we sample an i-colored ball, replace it and then *add in λ_i more balls of the same color*. Essentially by conditioning on the event that we add a ball, we have eliminated the randomness in the update while retaining the randomness in the sampling.

This latter Pólya urn can be represented by the stochastic addition matrix

$$\begin{pmatrix} \lambda_A & 0 \\ 0 & \lambda_B \end{pmatrix} \tag{1}$$

A very elegant result by Renlund (2010) provides a general expression for the long-term probability of seeing a A-colored ball.

Lemma 3 (Renlund (2010)) *Suppose we are given a Pólya urn with replacement matrix of the form*

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with a positive number of balls of each kind to start with. Assume that $a, b, c, d \geq 0$ and at least one entry is strictly positive. Then the limit of the fraction of balls of each type exists almost surely. The fraction p of A-colored balls can be characterized as follows:

- *If $a = d, c = b = 0$, then p tends towards a beta distribution.*
- *If not, then p tends towards a single point distribution x^*, where $x^* \in [0, 1]$ is a root of the quadratic polynomial*

  $$(c + d - a - b)x^2 + (a - 2c - d)x + c.$$

  If two such roots exist, then it is the one such that $f'(x^*) < 0$.

By limiting ourselves to the subsequence of events when some ball is added to the urn, and using the above general lemma characterizing the asymptotics of *deterministic* urn updates from Renlund (2010), we have the following lemma about the urn under this new assumption.

Lemma 4 *In the urn with addition matrix given above, the asymptotic probability of sampling a red ball is 1 if $\lambda_A > \lambda_B$ and is zero if $\lambda_B > \lambda_A$.*

Proof Invoking Lemma 3, we set the parameters $b = c = 0$, $a = \lambda_A$ and $d = \lambda_B$. The resulting quadratic polynomial is $(\lambda_B - \lambda_A)x^2 +$

$(\lambda_A - \lambda_B)x = 0$. This polynomial has two roots: $x = 0, 1$. The first derivative is $(\lambda_B - \lambda_A)(2x-1)$. If $\lambda_A > \lambda_B$, then this is negative for $x = 1$. Conversely, if $\lambda_B > \lambda_A$, then this is negative for $x = 0$. ■

Significance. In this scenario, the update process will view one region as having much more crime than the other, *even if crime rates are similar*. In particular, if region A has a crime rate of 10% and region B has a crime rate of 11%, the update process will settle on region B with probability 1. This is a classic "go with the winner" problem where feedback causes a runaway effect on the estimated probabilities.

3.4. Accounting for reported incidents

Now we consider what happens when we remove Assumption 3.4, i.e., we allow both discovered and reported incidents to be used as input to the urn model as is more usually the case in predictive policing systems. As discussed earlier, the weight terms w_d and w_r are used to weight discovered and reported crimes from a neighborhood, and so the total weight of crime incidents from (say) area A would be $w_d d_A + w_r r_A$ if it was visited, and $w_r r_A$ otherwise. This leads to the following urn replacement matrix:

$$\begin{pmatrix} w_d d_A + w_r r_A & w_r r_B \\ w_r r_A & w_d d_B + w_r r_B \end{pmatrix}$$

where, as in Section 3.3, we should interpret the entries of the matrix as expected values of a Bernoulli process.

Using the same trick as in Section 3.3, we can reinterpret the above matrix as a *deterministic* update process, and invoke Lemma 3 to understand the limiting behavior. The corresponding quadratic equation associated with this replacement urn is given by:

$$f(x) = w_d(d_B - d_A)x^2 + (w_d(d_A - d_B) - w_r(r_A + r_B))x + w_r r_A = 0$$

Let $R = w_r(r_A + r_B)$ be the total weight of reported incidents, and let us denote $w_d(d_B - d_A)$ by Δ_d, the weighted *differential* in discovered crime. We can then rewrite the above expression as:

$$f(x) = \Delta_d x^2 - (\Delta_d + R)x + w_r r_A = 0$$

We can now find the roots of $f(x) = 0$. These are given by

$$x = \frac{(\Delta_d + R) \pm \sqrt{(\Delta_d + R)^2 - 4\Delta_d w_r r_A}}{2\Delta_d}$$

which can be written as

$$x = \nu \pm \sqrt{\nu^2 - \frac{w_r r_A}{\Delta_d}}$$

where $\nu = \frac{1}{2} + \frac{R}{2\Delta_d}$. Taking the first derivative,

$$f'(x) = 2\Delta_d x - (\Delta_d + R)$$

and thus $f'(x) < 0$ when $x < \frac{1}{2} + \frac{R}{2\Delta_d}$. Therefore, by Lemma 3, the limiting fraction of "A-colored" balls in the urn is

$$x^* = \nu - \sqrt{\nu^2 - \frac{w_r r_A}{w_d(d_B - d_A)}} \tag{2}$$

3.4.1. Interpretation

We can interpret Equation (2) through a number of cases. Firstly, consider the case of *no feedback*. This corresponds to setting $w_d = 0$. In that case, the urn replacement matrix is fixed: regardless of which ball we draw, we always add r_A A-colored and r_B B-colored balls. Clearly, the limiting fraction of A-colored balls is $\frac{r_a}{r_a + r_b}$ and this is the answer we would expect given the crime reporting rates – we denote this fraction as λ^*.

We can rewrite Equation (2) in terms of λ^* by introducing a change of variable. Define $\kappa = R/\Delta_d$ which allows us to rewrite $\nu = (1 + \kappa)/2$. We can now rewrite Equation (2) as

$$x^* = \frac{1 + \kappa}{2} - \sqrt{\left(\frac{1 + \kappa}{2}\right)^2 - \lambda^* \kappa}. \tag{3}$$

The second term under the square root comes from noting that $w_r r_A/\Delta_d = r_A/(r_A + r_B) \cdot w_r(r_A + r_B)/\Delta_d$.

A first observation is that as $\lambda^* \to 0$, $x^* \to \lambda^*$. Similarly, as $\lambda^* \to 1$, $x^* \to \lambda^*$. In other words, if the crime rates between the neighborhoods are heavily skewed, this urn will converge to a good approximation of the correct answer.

For intermediate values of λ^*, we transform the equation as follows:

$$x^* = \frac{1+\kappa}{2} - \sqrt{\left(\frac{1+\kappa}{2}\right)^2 - \lambda^*\kappa}$$
$$= \frac{1+\kappa}{2}\left(1 - \sqrt{1 - \frac{\lambda^*\kappa}{\left(\frac{1+\kappa}{2}\right)^2}}\right)$$

which in the limit, as κ grows, can be expressed as

$$x^* = \frac{1+\kappa}{2}\left(\frac{\lambda^*\kappa}{2\left(\frac{1+\kappa}{2}\right)^2}\right)$$

by a binomial approximation, yielding

$$x^* = \lambda^*\frac{\kappa}{\kappa+1} = \lambda^*\frac{R}{R+\Delta_d}$$

Significance The limiting behavior of this urn is represented by x^*. How does this relate to the ideal limiting behavior λ^*? For $x^* \approx \lambda^*$, it must be that the ratio $R/(R+\lambda_d)$ is close to 1. This can happen in two ways. Either R must be very large, or Δ_d must be small. $R = w_r(r_a + r_b)$ which is bounded by 2. Thus, the only other option is to have Δ_d be very small. Recall that $\Delta_d = w_d(d_B - d_A)$. To make it small, we must either make w_d small, which corresponds to discounting the importance of discovered incidents (thus relying heavily on the distribution of reported incidents assumed to be correct by Assumption 3.3), or it must be that the discovered crime rates d_B and d_A are very similar. In other words, the only scenarios where feedback does *not* drive the outcome away from the true result are when we effective ignore feedback (by down-weighting the importance of discovered crime) or when the crime rates are similar enough for the feedback to not matter. However, it is precisely when crime rates are different that predictive policing is of value (because resources are then deployed differently). Thus, once again the urn model reveals problems (via simulation) in existing models for predictive policing.

3.5. Modifying the urn model to account for feedback

In order to learn the crime rate, we want the Pólya urn to contain balls in proportion to the relative probability of crime occurrence. As we have seen, a standard Pólya urn with stochastic update rates will converge to a distribution that has no relation to the true crime rates. Here, we present a simple change to the urn process which *does* guarantee that the urn proportion will converge to the ratio of replacement (i.e. crime) rates.

3.5.1. Discovered Incidents Only

Again, we first consider what happens if Assumption 3.4 is in place.

Consider the standard Pólya urn update rule: the probabilities λ_A and λ_B model the probability of an additional ball being added to the urn, *conditional* on a ball of the respective color having been sampled. This means that the probability of a ball being added is not λ_A, but $\lambda_A \frac{n_A^{(t)}}{n_A^{(t)}+n_B^{(t)}}$. As a result, the expected fraction of A-balls being added to the urn after one step of the process is $\frac{\lambda_A n_A}{\lambda_A n_A + \lambda_B n_B}$ instead of $\frac{\lambda_A}{\lambda_A+\lambda_B}$.

This immediately suggests a fix: instead of always adding the new balls, we *first sample another ball from the urn, and only add the new balls if the colors are different.* With this fix, the probability of adding a ball with label A is $\frac{n_A^{(t)}}{n_A^{(t)}+n_B^{(t)}}\lambda_A\frac{n_B^{(t)}}{n_A^{(t)}+n_B^{(t)}}$, while the probability of adding a ball with label B is $\frac{n_B^{(t)}}{n_A^{(t)}+n_B^{(t)}}\lambda_B\frac{n_A^{(t)}}{n_A^{(t)}+n_B^{(t)}}$. Crucially, these two expressions are proportional to λ_A and λ_B, except for a constant factor that is a function of the current state of the urn.

The intuition behind this fix is that if our decision procedure sends police to region A 90% of the time, we should not be surprised that discovered incidents in region A happen at a rate of nine to one, even if the crime rate is the same across both regions. In such a scenario, if we see a crime in region A (where police go 90% of the time), we should simply drop the incident record 90% of the time; analogously, in region B (where police only go 10% of the time), we drop the incident record 10% of the time.

One way to interpret our fix is as a form of *rejection sampling*: we are dropping sampled values according to some probability scheme to affect the statistic we are collecting. The importance-sampling analog to this scheme would be to use *weighted balls*, where the weight

of each ball is inversely proportional to be rate at which police are sent. Effectively, we want a scheme where as more police are sent, smaller weights are assigned to discovered incidents. But such a scheme is precisely the Thompson-Horvitz estimator, used in survey designs with unequal probability distributions (Horvitz and Thompson, 1952), and so we see that our proposal is a rejection-sampling variant of Thompson-Horvitz estimation.

3.5.2. Reported and Discovered Incidents

Now we consider what happens if there are both discovered and reported incidents. Intuitively, we want to correct for the runaway feedback caused by the discovered incidents, but not over-correct for the reported incidents, which don't suffer from the issue. Recall that the replacement matrix is:

$$\begin{pmatrix} w_d d_A + w_r r_A & w_r r_B \\ w_r r_A & w_d d_B + w_r r_B \end{pmatrix}$$

Suppose that Assumption 3.3 is in place and recall that $w_d + w_r = 1$. Then this replacement matrix is:

$$\begin{pmatrix} \lambda_A & w_r \lambda_B \\ w_r \lambda_A & \lambda_B \end{pmatrix} =$$

$$\begin{pmatrix} w_d d_A & 0 \\ 0 & w_d d_B \end{pmatrix} + \begin{pmatrix} w_r r_A & w_r r_B \\ w_r r_A & w_r r_B \end{pmatrix}$$

Note that the first matrix represents the discovered replacement and the second represents the reported replacement. From the previous section, we know how to modify the discovery replacement matrix so that the feedback effect is mitigated. We first apply that same technique here, *but only to the discovered incidents.* As before, doing this ensures that the replacement contributes (in expectation) exactly $w_d d_A$ to the urn when visiting A, and $w_d d_B$ when visiting B.

But what about the reported incidents? If we add them as is (i.e as given by the second matrix), the total contribution in the case of an A-visit is $w_d d_A + w_r r_A + w_r r_B$. Again, invoking Assumption 3.3, the total contribution becomes $\lambda_A + w_r \lambda_B$ (with a similar expression for a B-visit). Unfortunately, this expression leads to the urn converging to an incorrect rate, ultimately because the contribution to the region not visited has been incorrectly down-weighted. The fix is simple: we remove the down-weighting of the reported incidents in the neighborhood where police were not deployed. The resulting replacement matrix, in expectation, is:

$$\begin{pmatrix} w_d d_A + w_r r_A & r_B \\ r_A & w_d d_B + w_r r_B \end{pmatrix}$$

and we apply our earlier fix to any *discovered* data. This ensures that in expectation, the contribution to the urn *regardless* of whether A or B is visited is λ_A A-balls and λ_B B-balls, as desired.

4. Evaluating the urn model

In this section, we will focus on validating the existence of the feedback loop problem experimentally within our urn model. Code for our urn simulations can be found at https://github.com/algofairness/runaway-feedback-loops-src.

4.1. Observational decay

Thus far, our urn models have captured some key elements of the model used by `PredPol`– the idea of differential crime rates as well as the updates based on discovered and reported incidents. `PredPol` also includes a notion of *limited memory*, both by incorporating time decay into crime aftershocks, and by using a limited time window for training. We model limited memory in the urn setting by adding a simple notion of decay. After every round, each ball disappears from the urn independently with a fixed probability p_d. This can be thought of as a relaxation of Assumption 3.2. Varying p_d is analogous to varying the size of the `PredPol` training window.

4.2. Illustrating runaway feedback in urns

To the best of our knowledge there is no theoretical characterization of the asymptotic distributions in this full model once the notion of decay is included. We present empirical evidence illustrating the problems with using this model to learn crime rates. In our experiments, $p_d = 0.01$.

Using the Lum and Isaac (2016) data, we consider a two neighborhood police deployment scenario using, first, the two regions of Oakland with the most historical incident data ($Top1$

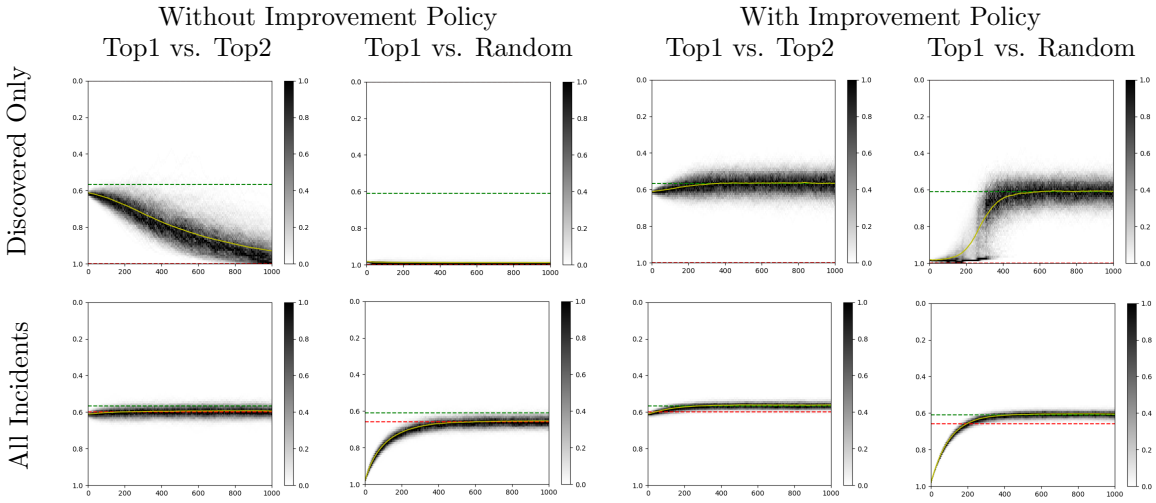

Figure 1: The distribution over 1000 days versus percentage of balls from region $Top1$ in the urn over 1000 runs. A police force deployed based on the underlying crime rates would send 56.7% of the force to $Top1$ instead of $Top2$ and 61.0% of the force to $Top1$ instead of *Random* (the green line shown). Top row (discovered incidents only): both charts (left) converge to sending 100% of the force to $Top1$, while with the improvement policy (right) the charts appear to converge to the correct crime rates. Bottom row (all incidents, equally weighted): both charts (left) converge to the incorrect rate (red line), while with the improvement policy (right) the charts appear to converge correctly to the true crime rates.

and $Top2$) and, second, the Oakland neighborhood with the most incidents as compared to a randomly chosen region with many fewer incidents (*Random*). We simulate the effect of the historical incident data on the prior for the system by determining the number of balls for each region in our urn based on the past number of incidents. We use the full number of incidents (609, 379, and 7 for regions $Top1$, $Top2$ and *Random* respectively) as the starting number of balls in the urn from each region. The urns are then updated based on the estimated number of daily drug use incidents, i.e., $\lambda_{Top1} = 3.69$, $\lambda_{Top2} = 2.82$, and $\lambda_{Random} = 2.36$.

4.2.1. Discovered Incidents Only

First, we begin with Assumption 3.4 in place stating that we'll only consider discovered incident data (i.e., $w_r = 0$). This allows us to isolate the part of the data that causes the feedback loop in order to examine its effect.

The results, shown in the top left of Figure 1, demonstrate that even if police sent to a neighborhood discover crime incidents according to the true crime rate (Assumption 3.3), the urn model will converge to *only* sending police to the neighborhood with the most crime. This replicates (within our urn model) the feedback loop problems with PredPol found by Lum and Isaac (2016). Recall, from Lemma 4, that skew occurs even if the difference in crime rates between the two neighborhoods is small. Note that while we included a notion of decay in our urn model in order to more closely model PredPol, we found similar results under the urn model without decay.

4.2.2. Discovered and Reported Incidents

Now, considering both reported and discovered incident data, we repeat the experiments. Again, we'll assume that both discovered and reported incidents are reported at the underlying true

rate (Assumption 3.3), and we'll assume that these incidents are equally weighted, i.e., that $w_d = w_r = 0.5$. The results shown in the bottom left of Figure 1 show that while the error in police deployment is not as great as if only discovered incidents are used, the urn still does not converge to the correct rate. Here, it's important to note the strength of the assumption that incidents are reported at the true underlying rate and not influenced by police deployment - we suspect that this assumption helps this convergence to be closer to (though still not the same as) the correct rate.

4.3. Evaluating the modified urn

Using this improvement policy to determine when to replace balls, we can now determine if the urns can learn the true crime rate despite the issue of observational bias. Again, using the estimated daily drug usage per region as the underlying true crime rate and the historical incident data as the prior for the urn color distribution, we simulate the urn's ability to find the relative crime rates in two regions, the $Top1$ and $Top2$ incident regions and a *Random* region. As shown in the right of Figure 1, urns under this improvement policy converge to a distribution of colors that represents the true crime rate, whether using only discovered incidents or both discovered and reported incidents.

5. Fixing PredPol

5.1. Modifying PredPol in a black-box manner

The urn models we explore provide a justification for the observed feedback loop failures of PredPol. But can we remedy PredPol itself using the improvements described in Section 3.5? We first demonstrate how asymmetric feedback affects PredPol by simulating the decisions a precinct might take after running it. We run PredPol's prediction model (using the Lum and Isaac (2016) data and implementation), trained on Oakland historical crime data, and generate crime according to the drug usage rates described above.

At each simulation day, PredPol trains on the previous 180 days of incident data, and produces predicted crime rates r_A and r_B. The decision of where to send police is made probabilistically, by a Bernoulli trial with $p = r_A(r_A+r_B)^{-1}$. This models the targeting effect of sending more police where more crime is expected, simulating a typical use of PredPol (Mohler et al., 2015).

To counteract the effects of the feedback, we can employ the same strategy as in Section 3.5. The key insight is that we need only filter the inputs presented to PredPol rather than trying to modify its internal workings. Specifically, once we obtain crime report data from the system, we conduct another Bernoulli trial with $p = r_O(r_A + r_B)^{-1}$, where r_O is the predicted rate of the district we did *not* police that day, and *only add the incidents to the training set if the trial succeeds.* In other words, the more likely it is that police are sent to a given district, the less likely it is that we should incorporate those discovered incidents.

5.2. Evaluating the PredPol simulation and its repair

Simulating the effects of PredPol on policing as described above, both before and after our improvement policy is applied, we compare the policing rates of region $Top1$ to $Top2$ and $Top1$ to *Random* as before. Each simulation is repeated 300 times and run for one year. As can be seen in the top row of Figure 2, regular PredPol rates fluctuate wildly over different runs, and do not converge to the appropriate crime rates (marked with the red dashed line). However, when the inputs to PredPol are changed according to our improvement policy, PredPol's prediction rates appear to fluctuate around the correct crime ratio. Note that the process is still quite noisy, a further indication that PredPol generates crime rate predictions that are still somewhat unreliable.

In Section 3.5.2, we provided an analysis and correction for urn models based on more than only discovered incidents. We provide a similar analysis for the mixed case in PredPol, shown along the bottom row of Figure 2. Note that even with a large number of reported incidents, PredPol seems to remain susceptible to runaway feedback. When the correction mechanism of Section 3.5.2 is applied to the (discovered only) incidents, PredPol appears to converge to the appropriate crime rate predictions.

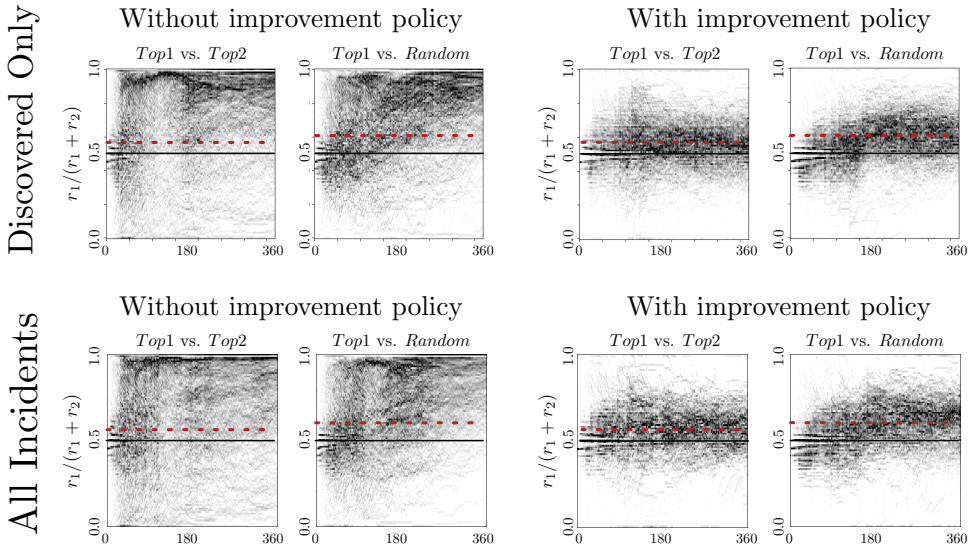

Figure 2: PredPol's relative deployment to region *Top*1 versus *Top*2 or *Random*. Along the top row, we use the model which only accounts for discovered incidents (those based on police having been deployed to an area). Along the bottom row, we use the model which accounts for both discovered and reported incidents. Left: PredPol operating as usual. Right: discovered incident entries modified using our improvement policy. Police deployment based on underlying crime rates would send 56.7% of the force to *Top*1 instead of *Top*2 and 61.0% of the force to *Top*1 instead of *Random*. These correct crime rates (indicated with a dashed red line) appear to be what PredPol converges to under the improvement policy.

6. Discussion and Limitations

In this paper we show that urn models can be used to formally model predictive policing as well as indicate remedies for problems with feedback. We demonstrate this both formally and empirically. Our solution also suggests a black-box method to counteract runaway feedback in predictive policing by appropriately filtering the inputs fed to the system.

Our results are not just a qualitative indicator of problems with feedback. They also indicate exactly how the problem of runaway feedback can be exacerbated: specifically as crime rates vary between regions and as the model relies more and more on discovered incident reports. Our results also indicate that if crime rates are more or less the same between regions, then the problem of feedback is much less, and it might be possible to generate reasonable predictions without explicitly countering feedback loops (though the results will still be inaccurate).

There are many avenues that our analysis does not yet explore. Firstly, while we expect that our solution generalizes to multiple regions (and indeed the problems with feedback remain exactly the same), there might be technical difficulties in working with the much smaller probabilities we will encounter. As an abstraction of predictive policing, an urn model suffices to capture feedback issues, but does not account for potentially richer predictive systems that might use other information (for example demographics) to make predictions. Another interesting dimension that is unexplored is the fact that different types of crime might have different reporting and discovery profiles, and might interact with each other in a predictive model in complex ways.

One of the most crucial assumptions we make (and one that in fact is sympathetic to current predictive policing models) is that reported and discovered incident rates track the true crime rates. There is considerable evidence that crime

reporting is noisy and skewed by area and by type of crime (Bialik, 2016). Once we remove that assumption, the analysis becomes more complicated, and while the problems of runaway feedback remain, the solutions might not continue to work. In this case, we would require better models to describe how crime rates manifest themselves in terms of reported and discovered incidents.

7. Acknowledgements

This paper would not have been possible without Kristian Lum and William Isaac's generosity in sharing the code and data developed for (Lum and Isaac, 2016). Many thanks!

References

Richard A. Berk and Justin Bleich. Statistical procedures for forecasting criminal behavior. *Criminology & Public Policy*, 12(3):513–544, 2013. ISSN 1745-9133. doi: 10.1111/1745-9133.12047. URL http://dx.doi.org/10.1111/1745-9133.12047.

Carl Bialik. How to make sense of conflicting, confusing and misleading crime statistics. https://fivethirtyeight.com/features/how-to-make-sense-of-conflicting-confusing-and-misleading-crime-statistics/, 2016. Visited 10/7/2017.

Ido Erev and Alvin E Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American economic review*, pages 848–881, 1998.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Kristian Lum and William Isaac. To predict and serve? *Significance*, pages 14 – 18, October 2016.

Hosam M Mahmoud and Basile Morcrette. Exactly solvable balanced tenable urns with random entries via the analytic methodology. *Discrete Mathematics & Theoretical Computer Science*, 2012.

George O. Mohler, Martin B. Short, Sean Malinowski, Mark Johnson, George E. Tita, Andrea L. Bertozzi, and P. Jeffrey Brantingham. Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association*, 110(512):1399 – 1411, 2015.

Inc. NorthPointe. Compas. http://www.northpointeinc.com/files/downloads/FAQ_Document.pdf, 2012.

Robin Pemantle. A survey of random processes with reinforcement. *Probab. Surveys*, 4:1–79, 2007. doi: 10.1214/07-PS094. URL http://dx.doi.org/10.1214/07-PS094.

Walt L Perry. *Predictive policing: The role of crime forecasting in law enforcement operations*. Rand Corporation, 2013.

Henrik Renlund. Generalized pólya urns via stochastic approximation. *arXiv preprint arXiv:1002.3716*, 2010.

U.S. DoJ via ICPSR. National crime victimization survey, 2014. United States Department of Justice. Office of Justice Programs. Bureau of Justice Statistics. [Distributor]: Inter-university Consortium for Political and Social Research (ICPSR), Aug. 27, 2015. URL http://doi.org/10.3886/ICPSR36142.v1.

U.S. HHS via ICPSR. National survey on drug use and health, 2011. United States Department of Health and Human Services. Substance Abuse and Mental Health Services Administration. Center for Behavioral Health Statistics and Quality. [Distributor]: Inter-university Consortium for Political and Social Research (ICPSR), 2015. URL http://doi.org/10.3886/ICPSR34481.v4.