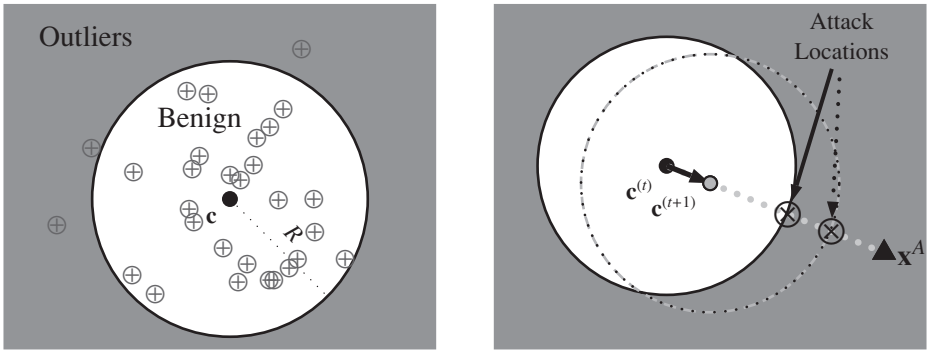
4 Attacking a Hypersphere Learner

In the second part of this book, we elaborate on *Causative* attacks, in which an adversary actively mistrains a learner by influencing the training data. We begin in this chapter by considering a simple adversarial learning game that can be theoretically analyzed. In particular, we examine the effect of malicious data in the learning task of anomaly (or outlier) detection. Anomaly detectors are often employed for identifying novel malicious activities such as sending virus-laden email or misusing network-based resources. Because anomaly detectors often serve a role as a component of learning-based detection systems, they are a probable target for attacks. Here we analyze potential attacks specifically against hypersphere-based anomaly detectors, for which a learned hypersphere is used to define the region of normal data and all data that lies outside of this hypersphere's boundary are considered to be anomalous. Hypersphere detectors are used for anomaly detection because they provide an intuitive notion for capturing a subspace of normal points. These detectors are simple to train, and learning algorithms for hypersphere detectors can be kernelized, that is implicitly extended into higher dimensional spaces via a kernel function (Forrest et al. 1996; Rieck & Laskov 2006; Rieck & Laskov 2007; Wang & Stolfo 2004; Wang et al. 2006; Warrender et al. 1999). For our purposes in this chapter, hypersphere models provide a theoretical basis for understanding the types of attacks that can occur and their potential impact in a variety of different settings. The results we present in this chapter provide intriguing insights into the threat of causative attacks. Then, in Chapter 5 and 6, we proceed to describe practical studies of causative attacks motivated by real-world applications of machine learning algorithms.

The topic of hypersphere poisoning first arose in designing virus and intrusion detection systems for which anomaly detectors (including hypersphere detectors) have been used to identify abnormal emails or network packets, and therefor are targets for attacks. This line of work sought to investigate the vulnerability of proposed learning algorithms to adversarial contamination. The threat of an adversary systematically misleading an outlier detector led to the construction of a theoretical model for analyzing the impact of contamination. Nelson (2005) and Nelson & Joseph (2006) first analyzed a simple algorithm for anomaly detection based on bounding the normal data in a mean-centered hypersphere of fixed radius as depicted in Figure 4.1(a). We summarize the results of that work in Sections 4.3 and 4.4. This analysis was then substantially extended by Kloft & Laskov (2010, 2012), whose work we summarize in Sections 4.5 and 4.6.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press



(a) Hypersphere Outlier Detection (b) Attack on a Hypersphere

Figure 4.1 Depictions of the concept of hypersphere outlier detection and the vulnerability of naive approaches. **(a)** A bounding hypersphere centered at $\mathbf{c}$ of fixed radius R is used to encapsulate the empirical support of a distribution by excluding outliers beyond its boundary. Samples from the "normal" distribution are indicated by $\oplus$'s with three outliers on the exterior of the hypersphere. **(b)** An attack against a hypersphere outlier detector that shifts the detector's "normal" region toward the attacker's goal $\mathbf{x}^A$. It will take several iterations of attacks to sufficiently shift the hypersphere before it encompasses $\mathbf{x}^A$ and classifies it as benign.

The novelty detection learning algorithm considered throughout this chapter is a mean-centered hypersphere of fixed radius R. For this basic model for novelty detection, we analyze a contamination scenario whereby the attacker poisons the learning algorithm to subvert the learner's ability to adapt to a tool the adversary uses to accomplish its objective. The specific scenario we consider is that the adversary wants the novelty detector to misclassify a malicious target point, $\mathbf{x}^A$, as a normal instance. However, the initial detector would correctly classify $\mathbf{x}^A$ as malicious so the adversary must manipulate the learner to achieve its objective. Initially, the attacker's target point, $\mathbf{x}^A$, is located a distance D_R radii from the side of the hypersphere (or a total distance of $R\,(D_R + 1)$ from the initial center). Further, it is assumed that the initial hypersphere was already trained using N initial benign data points, and the adversary has M total attack points it can deploy during the attack, which takes place over the course of T retraining iterations of the hypersphere model. Analyzing this simple attack scenario yields a deeper understanding into the impact of data contamination on learning agents and quantifies the relationship between the attacker's effort (i.e., M, the number of attack points used by the attacker) and the attacker's impact (i.e., the number of radii, D_R, by which the hypersphere is shifted).

4.1 Causative Attacks on Hypersphere Detectors

Learning bounding hyperspheres is a basic technique for anomaly detection that can be accomplished by learning a hypersphere centered at the empirical mean of a training set or that encloses the (majority of the) training data (e.g., see Shawe-Taylor & Cristianini 2004, Chapter 5). These novelty detection models classify all data that lie within the

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

bounding hypersphere as *normal* ("−") and all other data as *abnormal* ("+"). A simple version of this detector uses a mean-centered hypersphere of fixed radius R to bound the support of the underlying distribution as depicted in Figure 4.1(a). Such a detector is trained by averaging the training data, $\left\{\mathbf{x}^{(\ell)}\right\}$, to estimate the centroid as $\mathbf{c} = \sum_{\ell=1}^{N} \mathbf{x}^{(\ell)}$, and it classifies subsequent queries $\mathbf{x}$ as

$$f_{\mathbf{c},R}(\mathbf{x}) = \begin{cases} "+", & \text{if } \|\mathbf{x} - \mathbf{c}\| > R \\ "-", & \text{otherwise} \end{cases},$$

where we use $f_{\mathbf{c},R}$ to denote the classification function corresponding to the hypersphere centered at $\mathbf{c}$ with a radius R. Because we are considering a sequence of detectors with a fixed radius R but a changing centroid, we use the notation f_t to denote the t^{th} such detector with centroid $\mathbf{c}^{(t)}$.

One can imagine several situations in which a malicious user wants to attack such an outlier detection algorithm. For example, an adversary may be searching for malicious points that erroneously lie within the hypersphere, or it could try to mislead the hypersphere by tampering with its training data. Here we consider a *Targeted Causative Integrity* attack on the simple mean-centered hypersphere outlier detector described earlier. This attack takes place over the course of T retraining iterations. In this attack, the goal of the attacker is to cause the hypersphere to have a final centroid, $\mathbf{c}^{(T)}$, that *incorrectly* classifies a specific attack point $\mathbf{x}^A$ as normal, making this a *Targeted Integrity* attack. We assume that, prior to the attack, the target $\mathbf{x}^A$ is correctly classified by the detector (i.e., $f_0\left(\mathbf{x}^A\right) = "+"$) and that the attacker does not want to modify $\mathbf{x}^A$, but rather wants to mistrain the learner so that, after the T retraining iterations, its objective is fulfilled (i.e., $f_T\left(\mathbf{x}^A\right) = "-"$). This is a *Repeated Causative* attack; see 3.6. To analyze this iterated game, we now specify the assumptions made about the learning process and attacker and then analyze optimal attacks on the detector in several different situations.

4.1.1 Learning Assumptions

This chapter focuses on iterated security games. As such, the learning algorithm discussed here is relatively simple: a novelty detector modeled as a mean-centered hypersphere of fixed radius R (possibly in a kernel space as discussed in Section 4.6.3) that contains most of the normal data. This outlier detector is trained from a corpus of data, which is initially assumed to be predominantly *benign* (perhaps the initial training set is vetted by human experts), and the initial (unattacked) centroid is $\mathbf{c}^{(0)}$. The radius R is typically selected to tightly bound the normal training data while having a low probability of false positives. Choosing the radius to meet these constraints is discussed in Shawe-Taylor & Cristianini (2004, Chapter 5), but for this work we assume the radius is specified a priori; i.e., it cannot be influenced by the adversary.

Importantly, as new data becomes available, it is used to periodically retrain the detector. We assume this new data remains unlabeled and is susceptible to adversarial contamination, but that the new data is filtered to limit this vulnerability by retraining *only on data points previously classified as normal*. In particular, we assume

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

the novelty detector uses *bootstrapping retraining*, in which the latest detector is used to remove outliers from the newly received data, sanitizing the data before it is used for retraining. Under this policy, data points classified as *normal* are always used in subsequent retraining while any point classified as an *outlier* is immediately discarded. Finally, we initially assume there is no *replacement* of data; i.e., new points are added to the training set but no points are ever removed from it, regardless of how the model subsequently changes. We relax this last assumption in Section 4.5 where we examine the effect of different policies for data replacement. Regardless, as a result of retraining, the hypersphere detector is described by a sequence of centroids $\left(\mathbf{c}^{(t)}\right)_{t=0}^{T}$ produced by each retraining iteration.

4.1.2 Attacker Assumptions

We also make specific assumptions about the attacker's knowledge and capabilities. Throughout this chapter, we generally assume the attacker is *omnipotent*; that is, it knows the learner's feature representation, it knows the training data and current state (parameters) of the learning algorithm (although in most of the attack variants it only needs the state), it knows the learning algorithm and its retraining policy, and it can precisely predict the impact its attack has on the detector. We also assume that the attacker has strong capabilities. We assume the attacker can insert arbitrary points in feature space (i.e., it is not hindered by limitations of the measurement map ξ or feature map ϕ discussed in Section 2.2.1) and that it can control *all* data once the attack commences, but it cannot alter the representations of existing points (including the initial training data and its target data point $\mathbf{x}^A$). We modify the assumption that the attacker can control *all* data in Sections 4.5 and 4.6.

Finally, we assume the attacker has the *goal* of causing the retrained classifier to misclassify its target point $\mathbf{x}^A$ as normal. We quantify the attacker's task in terms of three quantities: the distance D_R that the attacker must displace the hypersphere to accomplish its goal, the total number of points M that the attacker can use in the attack, and the total number of retraining iterations T during which the attack is executed. The quantity $D_R > 0$ is expressed relative to the hypersphere radius R as

$$D_R = \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R} - 1; \tag{4.1}$$

that is, the total number of radii by which the hypersphere must be shifted (in the direction of the attack) to achieve the attacker's goal.[1] The remaining quantities M and T are variables, and in this chapter, we explore bounds on them. First, in Section 4.3, we consider an attacker that only wants to use as few attack points as possible, and we investigate the minimum number of attack points M required to achieve its goal. Second, in Section 4.4, we consider an attacker that wants to affect its attack quickly, and we investigate both the minimum number of retraining iterations T required and the minimum number of attack points M required in a fixed execution time T.

[1] This displacement is nonpositive if the attack point $\mathbf{x}^A$ is initially classified as normal, in which case no attack is necessary.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

Under these assumptions, an intuitive sketch of an attack strategy emerges. Because the outlier detector is only retraining on points falling within this hypersphere, this attacker must displace its centroid by inserting *attack points* within the hypersphere. Moreover, since the centroid is a linear combination of its training data, the attacker can achieve an optimal displacement by judiciously inserting its attack points at the intersection of the hypersphere's boundary and the line segment from the mean of the hypersphere to its goal $\mathbf{x}^A$. This attack strategy is depicted in Figure 4.1(b). As we show in Section 4.2, this observation reduces the task of attack optimization to a one-dimensional problem since it is assumed that the attacker has exact knowledge of the desired direction. The only complexity in optimizing the attack remains in choosing the number of points to place at each iteration of the attack; this task is addressed throughout the remainder of this chapter.

4.1.3 Analytic Methodology

Before delving into the details of the attacks, we sketch our analytic method. Namely, in the subsequent sections, we provide bounds on the number of attack points $M^\star$ or the number of retraining iterations $T^\star$ required by an adversary to achieve a desired displacement D_R. To do so, we find attacks that optimally displace the hypersphere toward $\mathbf{x}^A$ and upper bound the displacement such an attack can achieve under a given size M and duration T. We then invert this upper bounds to create lower bounds on M and T based on the following lemma.

LEMMA 4.1 *For any functions $f : \mathbb{X} \to \mathbb{Y}$ and $g : \mathbb{X} \to \mathbb{Y}$ mapping $\mathbb{X} \subseteq \Re$ to $\mathbb{Y} \subseteq \Re$ such that g is strictly monotonically increasing on $\mathbb{X}$ (and hence invertible) with g everywhere upper bounding f (i.e., $\forall\, x \in \mathbb{X}, \quad f(x) \leq g(x)$), if, for any $y \in \mathbb{Y}$, $z \in f^{-1}(y) = \{x \in \mathbb{X} \mid f(x) = y\}$, then we have*

$$z \geq g^{-1}(y).$$

It follows that, when f is invertible, $f^{-1}(y) \geq g^{-1}(y)$.

Proof [due to Matthias Bussas] By the contrapositive, suppose $z < g^{-1}(y)$. Then, it follows that $f(z) \leq g(z) < g(g^{-1}(y)) = y$ where the strict inequality is due to the strict monotonicity of g. Thus, $z \notin f^{-1}(y)$. □

We use this result throughout this chapter to invert bounds on the maximum distance attainable by an optimal attack to bound $M^\star$ or $T^\star$. We now proceed with a formal description of attacks against these iteratively retrained hyperspheres.

4.2 Hypersphere Attack Description

As discussed earlier, the attacker's objective is to manipulate the retraining process and induce a sequence of hypersphere centroids $\left(\mathbf{c}^{(t)}\right)_{t=0}^{T}$ such that for some $T \in \mathfrak{N}_0$

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

it achieves its objective $f_T\left(\mathbf{x}^A\right) = \text{"}-\text{"}$, or rather

$$\left\|\mathbf{x}^A - \mathbf{c}^{(T)}\right\| \leq R, \tag{4.2}$$

for which we assume T is the first such iteration satisfying this condition. Alternatively, we can frame this problem as minimizing the squared distance between $\mathbf{x}^A$ and $\mathbf{c}^{(T)}$ relative to the squared radius of the hypersphere, allowing us to formulate the attacker's objective as

$$\min_{\mathbf{c}^{(T)}} \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(T)}\right\|^2}{R^2}. \tag{4.3}$$

Clearly, this objective is minimized by $\mathbf{c}^{(T)} = \mathbf{x}^A$, but the attacker cannot select $\mathbf{c}^{(T)}$ directly. Instead, it must choose a sequence of attack points that yield a sequence of centroids to ultimately achieve the desired effect as we detail below. However, first we further decompose the attacker's objective into a more convenient form.

To quantify the attack's progress, we introduce the *total relative displacement* achieved by an attack of t iterations. This vector is defined as the relative displacement of the centroid from its initial state to its position after t^{th} retraining iterations:

$$\mathbf{D}_t = \frac{\mathbf{c}^{(t)} - \mathbf{c}^{(0)}}{R}. \tag{4.4}$$

Using $\mathbf{D}_t$, we can rewrite the vector used in the adversary's optimization objective in Equation (4.3) as $\frac{\mathbf{x}^A - \mathbf{c}^{(t)}}{R} = \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{R} - \mathbf{D}_t$, which gives the following alternative optimization objective:

$$\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(t)}\right\|^2}{R^2} = \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|^2}{R^2} + \|\mathbf{D}_t\|^2 - 2\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R} \cdot \left(\mathbf{D}_t^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}\right).$$

The first term $\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|^2}{R^2}$ is constant with respect to the attack and can be discarded. The remaining two terms express that the displacement $\mathbf{D}_t$ should align with the vector $\mathbf{x}^A - \mathbf{c}^{(0)}$ (i.e., the desired displacement vector) while not becoming too large. This latter constraint reflects the fact that if the displacement vector were too large, the shifted hypersphere would *overshoot* the target point $\mathbf{x}^A$ and subsequently still classify it as an outlier. However, overshooting the target is an implementation detail that can be easily avoided by halting the attack once the objective is achieved. It is not necessary to explicitly model this behavior as part of the optimization because it is not a practical concern. Further, in this chapter, we study attacks that use the minimal effort to achieve their effort and do not overshoot the target.

Moreover, as suggested by the above expression, the final term is expressed as two factors. The first, $2\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R}$, is again constant with respect to $\mathbf{c}^{(t)}$. However, the second represents a particular geometric quantity. It is the length of the projection of $\mathbf{D}_t$ onto the desired attack direction $\mathbf{x}^A - \mathbf{c}^{(0)}$; i.e., $\text{proj}_{\mathbf{x}^A - \mathbf{c}^{(0)}}(\mathbf{D}_t) = \mathbf{D}_t^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$. By the

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

Cauchy-Schwarz inequality, we obtain the following pair of results:

$$\|\mathbf{D}_t\| \geq \left| \mathbf{D}_t^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \right|$$

$$\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(t)}\right\|}{R} \geq \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R} - \mathbf{D}_t^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}.$$

This confirms that accomplishing $\text{proj}_{\mathbf{x}^A - \mathbf{c}^{(0)}}(\mathbf{D}_t) \geq D_R$ is necessary for the attack to achieve the original objective in Equation (4.2). Further, maximizing this projection for a fixed attack budget will generally find attacks that best align with the desired attack direction and have maximum magnitude. Hence, to simplify the results of this chapter, we consider the following alternative objective, which seeks the largest possible alignment to the desired displacement vector without regard to the possibility of overshooting. This notion of the attack's progress was originally introduced by Kloft & Laskov (2012), but there was called the *relative displacement*.

DEFINITION 4.2 *Optimal Displacement:* An attack achieves an *optimal displacement* at the t^{th} retraining iteration if its relative displacement vector $\mathbf{D}_t$ has the highest alignment with the desired displacement vector $\frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{R}$. The attack objective is given by the *displacement alignment*

$$\rho(\mathbf{D}_t) = \mathbf{D}_t^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}. \tag{4.5}$$

The attacker seeks to find a $\mathbf{D}_t$ that maximizes $\rho(\mathbf{D}_t)$.

Optimizing this objective achieves the same optimal sequences as those from Equation 4.3 until the target is reached. In the remainder of this chapter, we study attacks that seek maximal displacement alignment.

Remark 4.3 When the t^{th} displacement vector perfectly aligns with the attack direction (i.e., it has no residual), then the displacement vector is given by $\mathbf{D}_t = \kappa \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$ for some $\kappa \in [0, D_R]$ and $\|\mathbf{D}_t\| = \kappa$. The progress of such an attack is given precisely in terms of κ as

$$\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(t)}\right\|}{R} = \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R} - \kappa.$$

This exact connection between the original goal and alignment objective is, in fact, attained in several of the attack scenarios discussed later.

4.2.1 Displacing the Centroid

Here we discuss the behavior of $\mathbf{c}^{(t)}$ and how the attacker can manipulate it to optimize Equation (4.3). By the bootstrap retraining policy, when an attacker adds a point, $\mathbf{a}^{(t)}$, to the t^{th} training set, the t^{th} centroid will be affected if the point is within a distance R of the current centroid; i.e., $\left\|\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right\| \leq R$. When that occurs, assuming that the attacker is the only source of new data during the attack, the attack point causes the

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

hypersphere to shift in the next iteration to a new centroid given by

$$\mathbf{c}^{(t)} = \frac{\mu_{t-1}}{\mu_{t-1}+1}\mathbf{c}^{(t-1)} + \frac{1}{\mu_{t-1}+1}\mathbf{a}^{(t)}, \tag{4.6}$$

which is a convex combination of the prior centroid $\mathbf{c}^{(t-1)}$ and newly introduced attack point $\mathbf{a}^{(t)}$. This combination is defined by coefficients computed in terms of μ_{t-1}, the total number of training points used to train $\mathbf{c}^{(t-1)}$. This term μ_{t-1} is analogous to the *mass* that supports the prior hypersphere since it determines how difficult that hypersphere is to shift. Under the assumptions that data points are never removed and that, during the attack, the attacker is solely responsible for new data points, $\mu_t = \mu_{t-1} + 1$ with an initial mass $\mu_0 = N$ given by the number of benign data points that were present before the attack began.

More generally, during the t^{th} retraining iteration, the attacker attacks the hypersphere with a set $\mathbb{A}^{(t)} = \left(\mathbf{a}^{(t,\ell)}\right)_{\ell=1}^{\alpha_t}$ consisting of α_t attack points all of which are within R of the current centroid. Again, we assume the attacker is the only source of new data during the attack. The number of data points in the t^{th} retraining iteration is now given by $\mu_t = \mu_{t-1} + \alpha_t$ or more generally as the *cumulative sum of mass*:

$$\mu_t = N + \sum_{\ell=1}^{t} \alpha_\ell. \tag{4.7}$$

Further, the new centroid is now given by the convex combination

$$\begin{aligned}\mathbf{c}^{(t)} &= \frac{\mu_{t-1}}{\mu_{t-1}+\alpha_t}\mathbf{c}^{(t-1)} + \frac{1}{\mu_{t-1}+\alpha_t}\sum_{\ell=1}^{\alpha_t}\mathbf{a}^{(t,\ell)} \\ &= \mathbf{c}^{(t-1)} + \frac{1}{\mu_t}\sum_{\ell=1}^{\alpha_t}\left(\mathbf{a}^{(t,\ell)} - \mathbf{c}^{(t-1)}\right),\end{aligned} \tag{4.8}$$

which leads to a natural notion of the *relative displacement* at the t^{th} iteration.

DEFINITION 4.4 The *relative displacement* at the t^{th} retraining iteration is defined as the displacement vector of the hypersphere centroid from the $(t-1)^{\text{th}}$ to the t^{th} iteration relative to the fixed radius R of the hypersphere. This vector is given by

$$\mathbf{r}_t \triangleq \frac{\mathbf{c}^{(t)} - \mathbf{c}^{(t-1)}}{R} = \frac{1}{R\cdot\mu_t}\sum_{\ell=1}^{\alpha_t}\left(\mathbf{a}^{(t,\ell)} - \mathbf{c}^{(t-1)}\right).$$

Further, the *total relative displacement* can be expressed as the sum of the attack's relative displacements: $\mathbf{D}_T = \sum_{t=1}^{T}\mathbf{r}_t$.

Remark 4.5 A deeper insight into the nature of this problem is revealed by Equation (4.8). In particular, it shows that the change in the mean after T iterations of the attack relative to the radius of hypersphere R is a sum of "cumulatively penalized gains." That is, the contribution of the t^{th} iteration of the attack is weighed down by the sum of the *mass* used in all iterations up to and including the current iteration.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

From the fact that $\left\|\mathbf{a}^{(t,\ell)} - \mathbf{c}^{(t)}\right\| \leq R$ for all attack points, we can use the generalized triangle inequality to obtain our first bound:

$$\|\mathbf{r}_t\| = \frac{1}{R \cdot \mu_t} \left\| \sum_{\ell=1}^{\alpha_t} \left(\mathbf{a}^{(t,\ell)} - \mathbf{c}^{(t-1)}\right) \right\| \leq \frac{\alpha_t}{\mu_t} \leq 1,$$

since $\alpha_t \leq \mu_t$. This leads to the following theorem and a general (albeit, weak) bound on the effort required by the adversary:

THEOREM 4.6 *The total relative displacement between $\mathbf{c}^{(T)}$ and $\mathbf{c}^{(0)}$ in T retraining iterations has a norm of at most T (i.e., $\|\mathbf{D}_T\| \leq T$) and a displacement alignment of $\rho(\mathbf{D}_T) \leq T$. Therefor, to achieve the desired total relative displacement of D_R, a successful attack must have at least $T \geq D_R$ attack iterations.*

Proof. The bound on the norm follows from the generalized triangle inequality and the fact that for all t, $\|\mathbf{r}_t\| \leq 1$. Similarly, the bound on $\rho(\cdot)$ follows from the following application of the Cauchy-Schwarz inequality:

$$\begin{aligned}\rho(\mathbf{D}_T) = \mathbf{D}_T^\top \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} &= \frac{1}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \sum_{t=1}^{T} \mathbf{r}_t^\top \left(\mathbf{x}^A - \mathbf{c}^{(0)}\right) \\ &\leq \frac{1}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \sum_{t=1}^{T} \|\mathbf{r}_t\| \left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\| \leq T\end{aligned}$$

□

Ultimately, the attacker's goal is create a sequence of attack points (i.e., a set of attack points $\mathbb{A}^{(t)} = \left(\mathbf{a}^{(t,\ell)}\right)_{\ell=1}^{\alpha_t}$ at each attack iteration) such that the attacker's goal is satisfied. The following theorem states that the attacker can accomplish this in a greedy fashion by placing all its attack points at the point where the current hypersphere boundary intersects the line segment between the current centroid $\mathbf{c}^{(t-1)}$ and its goal point $\mathbf{x}^A$. Further, it shows that, when this greedy strategy is executed at every iteration, the resulting centroid at the t^{th} iteration follows the line segment between the initial centroid $\mathbf{c}^{(t-1)}$ and the attacker's goal point $\mathbf{x}^A$, gradually shifting toward its goal.

THEOREM 4.7 *For every attack sequence $\boldsymbol{\alpha} = (\alpha_t \in \mathfrak{N}_0)$ and for all $t \in \mathfrak{N}$, at the t^{th} iteration the set of attack vectors $\mathbb{A}^{(t)}$ that optimize $\rho(\cdot)$ according to Equation* (4.5) *consists of α_t copies of the vector $\mathbf{c}^{(t-1)} + R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$ and*

$$\mathbf{c}^{(t)} = \mathbf{c}^{(0)} + R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \cdot \sum_{\ell=1}^{t} \frac{\alpha_\ell}{\mu_\ell} \tag{4.9}$$

where μ_t is the cumulative sum of mass for the attack given by Equation (4.7).

Proof. The proof appears in Appendix B.1. □

This theorem shows that the optimal centroid at the t^{th} iteration can be computed in a greedy fashion only from the supplied parameters $\mathbf{c}^{(0)}$, R, $\mathbf{x}^A$, and $\boldsymbol{\alpha}$.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

COROLLARY 4.8 *For every attack sequence* $\boldsymbol{\alpha} = (\alpha_t \in \mathfrak{N}_0)$ *and for all* $T \in \mathfrak{N}$*, the total relative displacement achieved by an optimal attack following the attack sequence* $\boldsymbol{\alpha}$ *after* T *iterations is* $\mathbf{D}_T = \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \cdot \sum_{\ell=1}^{T} \frac{\alpha_\ell}{\mu_\ell}$*, which achieves a displacement alignment (Equation 4.5) of*

$$\rho\left(\mathbf{D}_T\right) = \sum_{\ell=1}^{T} \frac{\alpha_\ell}{\mu_\ell} \tag{4.10}$$

where μ_t *is the cumulative sum of mass for the attack given by Equation* (4.7) *and* D_R *is a parameter of the attack given by Equation* (4.1)*.*

Proof. The result for $\mathbf{D}_T$ follows directly from substituting the optimal centroid given by Equation (4.9) into Equation (4.4). The resulting displacement alignment follows from $\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|^2 = \left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)^\top \left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)$ and Equation 4.1. □

Importantly, under our assumptions, this theorem shows that the attacker's objective depends solely on the sequence $\boldsymbol{\alpha} = (\alpha_t)_{t=1}^T$; the actual attack vectors follow directly from its specification. The attacker can choose the elements of $\boldsymbol{\alpha}$; i.e., the number of attack points to employ at each iteration. Hence, we have reduced a multidimensional optimization problem to an optimization over a single sequence. In the next section, we formalize how the attacker can optimize this sequence based on the attack objective given in Equation (4.10).

However, before continuing, note that Equation (4.10) shows that the *success* of an attack at time t can be described solely as a function of this attack sequence. Moreover, since the optimal displacement vector $\mathbf{D}_T$ is a scalar multiple of the desired attack direction, $\mathbf{x}^A - \mathbf{c}^{(0)}$, its projection onto that direction has no residual component, and Remark 4.3 shows us that the progress of this attack is given by

$$\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(t)}\right\|}{R} = \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{R} - \sum_{\ell=1}^{T} \frac{\alpha_\ell}{\mu_\ell}.$$

The success of this attack in minimizing $\frac{\left\|\mathbf{x}^A - \mathbf{c}^{(t)}\right\|}{R}$ is determined completely by the attacker's choice of $\boldsymbol{\alpha}$, which it chooses so as to maximize $\sum_{\ell=1}^{T} \frac{\alpha_\ell}{\mu_\ell}$. We now proceed to formalize this setting by describing these attack sequences.

Remark 4.9 (Nontrivial Initial Attack) The astute reader will have noticed that the above results, including all derivations from Equation (4.8) onward, rely on the assumption that $\forall t \in \{1, \ldots, T\}$, $\mu_t > 0$. Equivalently, this requires that $\alpha_1 > 0$, which we assume throughout the remainder of this chapter—the *nontrivial initial attack assumption*. In fact, if the first nonzero attack occurs at the k^{th} iteration (i.e., $\alpha_t = 0$ for $t < k$ and $\alpha_k > 0$), the first $k - 1$ iterations can be discarded from the attack, since there is no adversarial impact on the classifier during this period. Further, the attack sequence of all zeros, $\boldsymbol{\alpha} = \mathbf{0}$, is the *trivial attack sequence* and need not be considered.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

4.2.2 Formal Description of the Attack

Having described the problem and the assumptions made under it, a formal analysis can be conducted to reveal optimal attack strategies. This formal analysis begins with a formalization of the problem setting and of the objective. We refer to the number of *attack points* used at the t^{th} retraining iteration as α_t and the optimal (under prescribed conditions) number of attack points to be used at the t^{th} iteration as $\alpha_t^\star$. We use $\mathfrak{N}$ to denote the natural numbers $1, 2, 3, \ldots$, $\mathfrak{N}_0$ to denote the non-negative integers $0, 1, 2, 3, \ldots$, and $\Re_{0+}$ to denote the non-negative reals. Unless specifically mentioned otherwise, $\alpha_t, \alpha_t^\star \in \mathfrak{N}_0$ although later in the text we consider sequences in the non-negative reals, which we differentiate notationally by using $\beta_t, \beta_t^\star \in \Re_{0+}$, respectively.

Along with α_t and β_t, we also define the space of possible attack sequences. Formally, we define $\mathcal{A}$ to be the space of all legitimate sequence of attack points; that is, $\mathcal{A} = \left\{ (\alpha_t)_{t=1} \;\middle|\; \forall t\ \alpha_t \in \mathfrak{N}_0 \right\}$ (in this space, any attack of a finite span is represented by concatenating an infinite trailing sequence of zeros). Similarly, we define the space of attacks of finite duration and limited size as $\mathcal{A}^{(M,T)} = \left\{ (\alpha_t)_{t=1}^T \;\middle|\; \forall t\ \alpha_t \in \mathfrak{N}_0 \ \wedge\ \sum_{t=1}^T \alpha_t \leq M \right\}$; the space of attacks of a finite duration, T, but unlimited size as $\mathcal{A}^{(\infty,T)}$; and the space of attacks of a limited total size, M, but unlimited duration as $\mathcal{A}^{(M,\infty)}$. Finally, the analogous continuous versions of these spaces are denoted by $\mathcal{B}$, $\mathcal{B}^{(\infty,T)}$, $\mathcal{B}^{(M,\infty)}$, and $\mathcal{B}^{(M,T)}$ and defined by replacing α_t with $\beta_t \in \Re_{0+}$ in the corresponding definitions of $\mathcal{A}$.

With the notion of an attack sequence, we now formalize the notion of optimal strategies by reexamining the objective of the attacker. The attacker wishes to maximize the displacement alignment $\rho(\cdot)$ as described in Definition 4.2, which was shown in Corollary 4.8 to depend solely on the attack sequence. The objective function is defined with respect to a given attack sequence $\boldsymbol{\alpha} \in \mathcal{A}$ according to Equation (4.10) as

$$D(\boldsymbol{\alpha}) = \sum_{t=1} \frac{\alpha_t}{\mu_t} = \sum_{t=1} \delta_t(\boldsymbol{\alpha}) \tag{4.11}$$

$$\delta_t(\boldsymbol{\alpha}) = \frac{\alpha_t}{\mu_t}, \tag{4.12}$$

where $\mu_t = N + \sum_{\ell=1}^t \alpha_\ell$ from Equation (4.7) and the function $\delta_t(\cdot)$ assesses the *contribution* due to the t^{th} iteration of the attack, which depends on only the first t elements of the attack sequence. The goal of the attacker is to maximize this objective function $D(\cdot)$ with respect to constraints on the size and duration of the attack.

DEFINITION 4.10 *Optimality:* An attack sequence $\boldsymbol{\alpha}^\star \in \mathcal{A}^{(M,\infty)}$ against a hypersphere with N initial non-attack points is an optimal strategy that uses a total of M attack points if $\forall \boldsymbol{\alpha} \in \mathcal{A}^{(M,\infty)}$, $D(\boldsymbol{\alpha}) \leq D(\boldsymbol{\alpha}^\star)$. The optimal distance achieved by such a sequence is denoted by $D_N^\star(M, \infty)$, where ∞ here represents the infinite attack duration. This optimality can be achieved for the attacker by solving the following program for $\boldsymbol{\alpha}^\star$:

$$\begin{aligned} \boldsymbol{\alpha}^\star \in \operatorname{argmax}_{\boldsymbol{\alpha}} & D(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \boldsymbol{\alpha} \in \mathcal{A}^{(M,\infty)} \end{aligned} \tag{4.13}$$

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

Thus, $D_N^\star(M, \infty)$, the optimal distance achievable for any sequence in the space $\mathcal{A}^{(M,\infty)}$, is expressed in terms of the problem's parameters M and N; i.e., if $\boldsymbol{\alpha}^\star$ is an optimal strategy in $\mathcal{A}^{(M,\infty)}$, then $D(\boldsymbol{\alpha}^\star) = D_N^\star(M, \infty)$. Similarly for attacks constrained to a finite duration T in the space of sequences $\mathcal{A}^{(M,T)}$, we define the optimal achievable distance to be $D_N^\star(M, T)$, which we return to in Section 4.4.

4.2.3 Characteristics of Attack Sequences

To better understand our problem, we characterize its properties and those of (optimal) attack sequences. These properties provide the foundation for the further analysis of the problem.

4.2.3.1 Invariance to Empty Attack Iterations

We discuss the behavior of the attack distance $D(\cdot)$ from Equation 4.11 with respect to zero elements in the attack sequence. First we show that the attack distance $D(\cdot)$ is invariant to the insertion of a zero at the k^{th} position in the sequence (with $k > 1$ following Remark 4.9).

LEMMA 4.11 *For any $k > 1$, every sequence $\boldsymbol{\alpha} \in \mathcal{A}^{(M,\infty)}$ achieves an identical distance as the sequence $\boldsymbol{\alpha}'$ defined as*

$$\alpha'_t = \begin{cases} \alpha_t, & \text{if } t < k \\ 0, & \text{if } t = k \\ \alpha_{t-1}, & \text{if } t > k \end{cases};$$

i.e., $D(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha}')$.

Proof. First, $\delta_t(\boldsymbol{\alpha}') = \delta_t(\boldsymbol{\alpha})$ for $t < k$ since $\delta_t(\cdot)$ depends only on the first t elements of the sequence (see Equation 4.12). Second, $\delta_k(\boldsymbol{\alpha}') = 0$ from Equation (4.12). Third, $\delta_t(\boldsymbol{\alpha}') = \delta_{t-1}(\boldsymbol{\alpha})$ for $t > k$ since inserting a 0 at the k^{th} position does not affect the denominator μ_t in the definition of $\delta_t(\cdot)$ (see Equations (4.7) and (4.12)) and the numerators are shifted, accordingly. The distance achieved by the sequence $\boldsymbol{\alpha}'$ is

$$\begin{aligned} D(\boldsymbol{\alpha}') &= \sum_{t=1}^{k-1} \delta_t(\boldsymbol{\alpha}') + \delta_k(\boldsymbol{\alpha}') + \sum_{t=k+1} \delta_t(\boldsymbol{\alpha}') \\ &= \sum_{t=1}^{k-1} \delta_t(\boldsymbol{\alpha}) + \sum_{t=k+1} \delta_{t-1}(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha}). \end{aligned}$$

□

From this lemma, it follows that the insertion (or deletion) of zero elements ($\alpha_t = 0$) is irrelevant to the sequence's distance, and therefore all zeros can be removed in considering our notion of optimality. Intuitively, the distance achieved by an attack is not affected by the retraining iterations in which no adversarial data is used since, in this scenario, the adversarial data is the sole source of new data. This notion is captured by the following theorem:

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

THEOREM 4.12 *Every pair of sequences* $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}^{(M,\infty)}$ *with identical subsequences of nonzero elements—i.e.,* $(\alpha_t \mid \alpha_t > 0) = \left(\alpha'_t \;\middle|\; \alpha'_t > 0\right)$*—achieve the same distance:* $D(\boldsymbol{\alpha}) = D(\boldsymbol{\alpha}')$*; that is, the distance achieved by a sequence is independent of the number of zeros in the sequence and their placement.* $D(\boldsymbol{\alpha})$ *only depends on the subsequence of nonzero elements of* $\boldsymbol{\alpha}$*. As a consequence, any finite sequence can be reordered as its positive subsequence followed by a subsequence of all zeros, and the two achieve identical distances.*

Proof. Since the sequences $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ contain the same nonzero elements in the same order, one can transform $\boldsymbol{\alpha}$ to $\boldsymbol{\alpha}'$ by repeated applications of Lemma 4.11 to insert and delete zeros at the necessary positions in the sequence. Thus $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}'$, and all intermediate sequences used in this transformation have identical distances. □

It follows that zero elements can be arbitrarily inserted into or removed from any optimal attack sequence to form an equivalent optimal attack sequence with the same distance since zero elements neither add distance nor "weight" to the subsequent denominators. Theorem 4.12 allows us to disregard all zero elements in a sequence since they do not contribute to the effectiveness of the attack. Moreover, moving all zero elements to the end of the sequence corresponds to the notion that the attacker wants to minimize the time required for the attack since it does not benefit our attacker to prolong the attack. Finally, the fact that zero elements can be disregarded suggests the possibility of redefining $\alpha_t \in \mathfrak{N}$ rather than $\mathfrak{N}_0$.

4.2.3.2 Characteristics of Optimal Attack Sequences

Having shown that zero elements are irrelevant to our analysis, we now describe the properties of the nonzero elements in optimal attacks. To begin, in this attack formulation, there are no initial points supporting the hypersphere, so no matter how many points the attacker places in the first iteration, the same displacement is achieved. This is captured by the following lemma:

LEMMA 4.13 *For* $N = 0$*, the optimal initial attack iteration is given by* $\alpha_1^\star = 1$.

Proof. The contribution of the first attack iteration is given by $\delta_1(\boldsymbol{\alpha}) = \frac{\alpha_1}{\mu_1} = \frac{\alpha_1}{\alpha_1} = 1$. Hence, for $\alpha_1 \in \mathfrak{N}$ (we exclude the possibility that $\alpha_1 = 0$ in accordance with Remark 4.9), we have $\delta_1(\boldsymbol{\alpha}) = 1$, and since $\delta_t(\boldsymbol{\alpha}) = \frac{\alpha_t}{\alpha_1 + \sum_{\ell=2} \alpha_\ell}$ is strictly decreasing in α_1 for $t > 1$, the optimal integer solution for this first iteration is given by $\alpha_1^\star = 1$. □

Additionally, if the total attack capacity is greater than one attack point and the attacker has the ability to distribute its attack over more than one retraining iteration, it is beneficial for it to do so; i.e., attacks that concentrate all their attack points in a single attack iteration are nonoptimal. This is captured by Lemma B.2 provided in Appendix B.3.

Further, the notion of cumulatively penalized gains from Remark 4.5 is crucial. There are two *forces* at work here. On the one hand, placing many points (large α_t) during iteration t improves the contribution of the term $\delta_t(\boldsymbol{\alpha})$ to the overall distance $D(\boldsymbol{\alpha})$. On

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

the other hand, a large α_t will be detrimental to subsequent terms since it will increase the size of the denominator μ_τ in the contributions $\delta_\tau(\boldsymbol{\alpha})$ for $\tau > t$. This effect can be likened to having the mean of the points becoming heavier (harder to move) as more points are utilized. Intuitively, one does not want to place too much *weight* too quickly as it will cause the mean to become too *heavy* toward the end of the attack, making the latter efforts less effective. This suggests that any optimal attack sequence should be monotonically nondecreasing, which we prove in the following theorem:

THEOREM 4.14 *For any optimal sequence of attack points, $\boldsymbol{\alpha}^\star \in \mathcal{A}^{(M,\infty)}$, every subsequence of nonzero elements of $\boldsymbol{\alpha}^\star$ must be monotonically nondecreasing; that is, if $\mathbb{I}^{(nz)} = \left\{ i_1, i_2, \ldots \;\middle|\; \forall\, k \quad \alpha^\star_{i_k} > 0 \right\}$ is a set of indexes corresponding to nonzero elements of $\boldsymbol{\alpha}^\star$, then $\forall\, i, j \in \mathbb{I}^{(nz)} \quad i \leq j \;\Leftrightarrow\; \alpha^\star_i \leq \alpha^\star_j$.*

Proof. The proof appears in Appendix B.2. □

Note that Theorem 4.14 does not require strict monotonicity, which makes it consistent with Theorem 4.12.

While it has been shown that any optimal attack sequence should be monotonically nondecreasing in magnitude, the intuition that the mean becomes *heavier* as more points are utilized suggests more than just monotonicity. In fact, this notion will lead us to an optimal solution in Section 4.3.1.

4.2.3.3 Behavior of Optimal Attack Distances

While it can be difficult to describe optimal attacks, we can generally describe the behavior of the optimal displacement alignment (over all possible attacks) as functions of M and T. In particular, we would expect that as the number attack points available to the attacker, M, increases, the resulting optimal displacement alignment should increase. Similarly, as the attack duration T increases, we also expect the resulting optimal displacement alignment to increase. The following theorem shows that the function $D^\star_N(M, T)$ does, in fact, monotonically increase with respect to both M and T for any fixed $N \geq 0$.

THEOREM 4.15 *For all $N \in \mathfrak{N}_0$, the functions $D^\star_N(M, \infty)$ and $D^\star_N(M, T)$ (for any fixed $T > 0$) are strictly monotonically increasing with respect to $M \in \mathfrak{N}_0$ unless $N = 0$ and $T = 1$ in which case $D^\star_0(M, 1) = 1$ for all $M \in \mathfrak{N}_0$. Further, for any fixed $M > 0$, the function $D^\star_N(M, T)$ is strictly monotonically increasing with respect to $T \leq M$ and is constant for $T > M$; i.e., $D^\star_N(M, T) = D^\star_N(M, \infty)$ for any $T \geq M$.*

Proof. The proof appears in Appendix B.3. □

Note that, with respect to T, the function $D^\star_N(M, T)$ is constant for $T \geq M$ because the attacker must use at least one attack point in every gainful retraining iteration (see Section 4.2.3.1). The attacker gains no additional benefit from attacks that exceed M in duration.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

4.3 Optimal Unconstrained Attacks

We now present solutions to different variations of the hypersphere attack problem and find optimal attack strategies, as defined earlier. In this section, we explore attacks without any constraints on the attacker, and, in the following sections, we consider different constraints that make the attacks more realistic. For an unconstrained attacker, the strict monotonicity properties demonstrated in Theorem 4.15 suggest that an optimal sequence should use all M available attack points and space its points as uniformly as possible to maximally extend the attack duration, T, after discarding zero elements. Indeed, this is an optimal integer strategy for the optimization problem in Definition 4.10, which is proven in the following theorem:

THEOREM 4.16 *For $N \in \mathfrak{N}$, any optimal attack sequence, $\boldsymbol{\alpha}^\star \in \mathcal{A}^{(M,\infty)}$, must satisfy $\alpha_t^\star \in \{0, 1\}$ and $\sum_t \alpha_t^\star = M$; i.e., $\boldsymbol{\alpha}^\star$ must have exactly M ones. In particular, one such optimal sequence is $\mathbf{1}_M$, which is a sequence of M ones followed by zeros. Moreover, the optimal displacement achieved by any $\boldsymbol{\alpha}^\star \in \mathcal{A}^{(M,\infty)}$ is $D_N^\star(M, \infty) = h_{M+N} - h_N$ where $h_k = \sum_{\ell=1}^{k} \frac{1}{\ell}$ is the k^{th} harmonic number.*

Proof. The proof appears in Appendix B.4. □

As a result of this theorem, we have a tight upper bound on the effect of any attack that uses M attack points. While the harmonic numbers are computable, there is no closed-form formula to express them. However, using the fact that $h_{M+N} - h_N = \sum_{k=1}^{M} \frac{1}{k+N}$ is a series of a decreasing function in k, it is upper bounded by $\int_0^M \frac{dx}{x+N} = \ln\left(\frac{M+N}{N}\right)$ when $N > 0$ (Cormen, Leiserson, Rivest, & Stein 2001, Appendix A.2). Similarly, when $N = 0$, Cormen et al. (2001, Appendix A.2) show that $h_k \leq \ln(k) + 1$. Together, we have the following upper bound on the optimal displacement achieved by an attack with M points and no time limitations:

$$D_N^\star(M, \infty) \leq \begin{cases} \ln(M) + 1, & \text{if } N = 0 \\ \ln\left(\frac{M+N}{N}\right), & \text{if } N > 0 \end{cases}.$$

Since these upper bounds are strictly increasing functions in M, we apply Lemma 4.1 to invert the bounds and obtain the following lower bounds on the the number of attack points required to execute an attack that displaces the hypersphere by the desired relative displacement, D_R. These bounds are simply

$$M^\star \geq \begin{cases} \exp(D_R - 1), & \text{if } N = 0 \\ N(\exp(D_R) - 1), & \text{if } N > 0 \end{cases};$$

i.e., the effort required by the attacker to achieve its goal grows exponentially in D_R.

4.3.1 Optimal Unconstrained Attack: Stacking Blocks

The optimal strategy given by $\boldsymbol{\alpha}^\star = \mathbf{1}_M$ can alternatively be derived by transforming this problem into a center-of-mass problem. Recall that, in Remark 4.5, the distance

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

achieved by the attack was likened to a sum of cumulatively penalized gains. We can think of this as a sequence of contributions attributable to each iteration of the attack; that is, the t^{th} iteration of the attack contributes $\delta_t(\boldsymbol{\alpha}) = \frac{\alpha_t}{\sum_{\ell=1}^{t} \alpha_\ell}$ which is the "amount of weight" used at time t relative to the total weight used up to that time. This is analogous to a center-of-mass problem. In particular, if we model the attack points α_t as units of mass that are placed at a distance of R from the current center of mass $\mathbf{c}^{(t-1)}$, $\delta_t(\boldsymbol{\alpha})$ given by Equation (4.12) is the amount by which the center of mass $\mathbf{c}^{(t)}$ is shifted relative to R. Since viable attack points cannot be placed beyond distance R, this is analogous to placing a set of identical blocks below the current stack of blocks that was created at time $t-1$ such that the stack does not topple (the structure being stable corresponds to the constraint that viable attack points cannot be beyond the radius R). Since the attack is constrained to only place points at the boundary, this analogy only holds when the stacking is done optimally or some of the blocks are vertically grouped (corresponding to placing several attack points in a single iteration, a notion that will be revisited in Section 4.4). Figure 4.2 depicts the correspondence between attacks on mean-centered hyperspheres and the stacking of blocks extending beyond the edge of a table.

Having likened the attack strategy to a classical physics problem, the optimal strategy reemerges from the latter's solution. As is discussed in (Johnson 1955), the blocks can be optimally stacked by extending the first by $\frac{1}{2}$, the second by $\frac{1}{4}$, and the t^{th} by $\frac{1}{2t}$. The optimal integer strategy is given by placing a single point per iteration. Moreover, as is mentioned in Figure 4.2, since the blocks are of length $2R$, this optimal strategy achieves a displacement determined by the harmonic series $D_0^\star(M, \infty) = h_M = \sum_{t=1}^{M} \frac{1}{t}$. We arrive at a physical representation for the hypersphere attack and the corresponding optimal strategy that we derived in Theorem 4.16. (In fact, the single-block stacking strategy is not optimal if one allows more than one block per vertical layer, as per the multi-wide stacking problem (Hall 2005; Hohm, Egli, Gaehwiler, Bleuler, Feller, Frick, Huber, Karlsson, Lingenhag, Ruetimann, Sasse, Steiner, Stocker, & Zitzler 2007). However, due to the constraints of our problem, such stacking strategies do not correspond to realistic attacks as they would imply adding attack points outside of the hypersphere.

4.4 Imposing Time Constraints on the Attack

In the previous section, we showed that optimality was achieved by attacks that use at most one attack point at each retraining iteration. While this strategy achieves maximal possible displacement toward the attacker's target for any fixed attack budget M, it fails to capture the entire objective of the attacker. Namely, the goal of an attack is to achieve an objective (displace the mean a desired amount), but to do so within a minimal total attack duration T or with minimal effort (fewest possible points M). As the preceding analysis shows, the prescribed attack achieves maximal distance of $\approx \log M$ but does so in time $T = M$, and thus, such an attack only achieves a logarithmic effect in the time required to mount the attack, whereas Theorem 4.6 bounds the total displacement achieved linearly in T. This discrepancy between this upper bound and the actual effect achieved suggests that such an attack does not fully utilize the attacker's available resources; i.e., its attack budget, M. As such, we consider the case of an

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

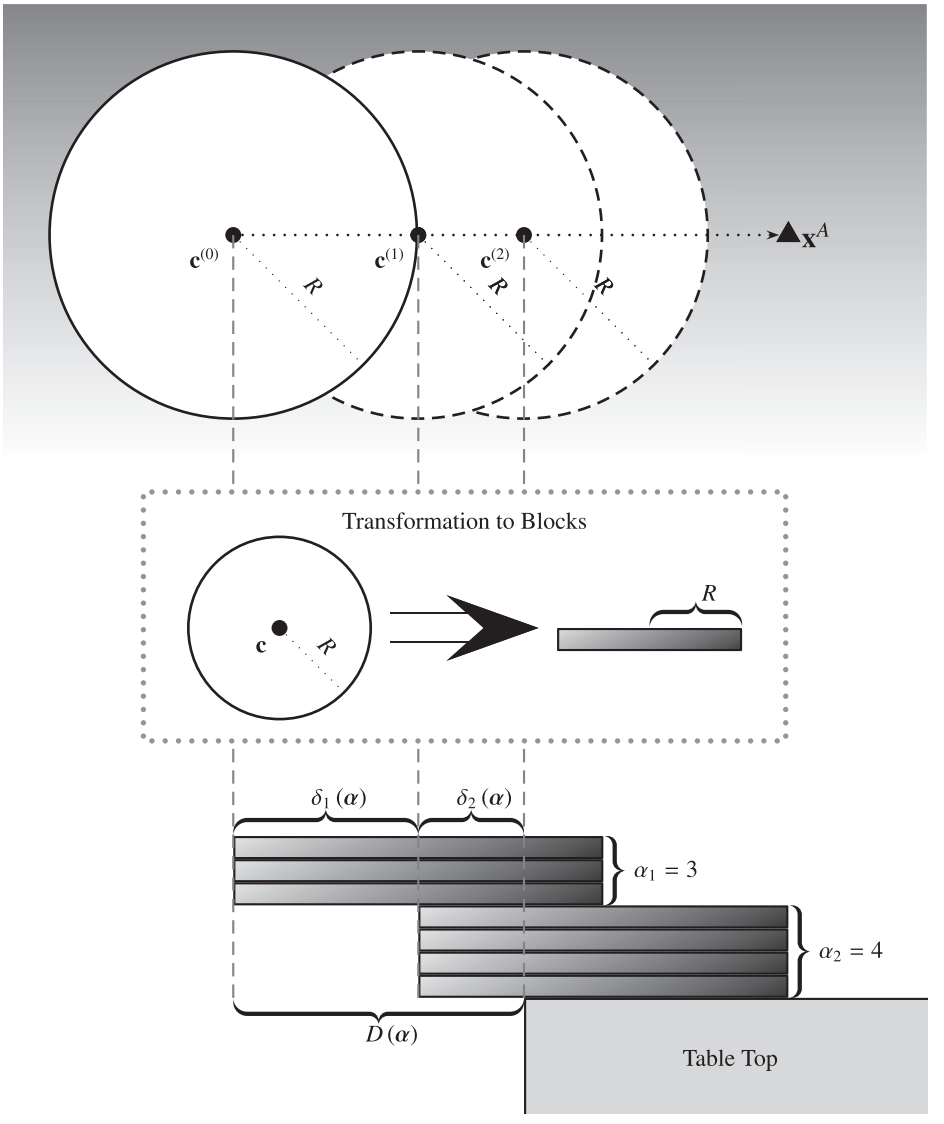


Figure 4.2 A figure depicting the physics analogy between the attack sequence $\boldsymbol{\alpha} = (\alpha_1 = 3, \alpha_2 = 4)$ and the concept of optimally stacking blocks on the edge of a table to extend beyond the edge of the table. From top to bottom, the original effect of the attack $\boldsymbol{\alpha}$ on the naive hypersphere outlier detector is transformed into the equivalent balancing problem. In this analogy, blocks of length $2R$ with a starting edge at $\mathbf{c}^{(t)}$ are equivalent to placing an attack point at the t retraining iteration of a hypersphere with mean $\mathbf{c}^{(t)}$ and radius R. This strange equivalence encapsulates the idea of a point of unit mass being placed at a distance R from the former mean. Vertical stacks can be interpreted as placing several points at time t, and time (oddly enough) flows down the blocks to the table. Also depicted are the contributions $\delta_1(\boldsymbol{\alpha})$ and $\delta_2(\boldsymbol{\alpha})$ along with their overall effect $D(\alpha_1, \alpha_2)$.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

attacker, who must execute its attack within T retraining iterations for some $T \in \mathfrak{N}_0$, with the more realistic assumption that $T \ll M$; i.e., the attack must occur in a small time window relative to the total size of the attack. This leads to the following notion of constrained optimality:

DEFINITION 4.17 *Constrained Optimality* An attack sequence $\boldsymbol{\alpha}^\star \in \mathcal{A}^{(M,T)}$ is considered optimal with respect to M total available attack points and a given duration T if $\forall \boldsymbol{\alpha} \in \mathcal{A}^{(M,T)} \quad D(\boldsymbol{\alpha}) \leq D(\boldsymbol{\alpha}^\star)$ and the optimal distance achieved by such a sequence is denoted by $D_0^\star(M, T)$. This optimality can be achieved by the attacker by solving the following program to find an optimal attack sequence $\boldsymbol{\alpha}^\star$:

$$\begin{aligned} \boldsymbol{\alpha}^\star \in \operatorname{argmax}_{\boldsymbol{\alpha}} & D(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \boldsymbol{\alpha} \in \mathcal{A}^{(M,T)} \end{aligned} \tag{4.14}$$

An equivalent formulation of constrained optimality would take a desired relative displacement D_R as an input and attempt to minimize the duration T required to achieve the desired distance with a total of M attack points. Similarly, another alternative would be to minimize M with respect to a fixed D_R and T. However, Equation (4.14) is a natural way to think about this optimization in terms of the attacker's goal. In the remainder of this section, we derive bounds that can be achieved for this constrained problem.

Before we continue, note that, when $M \geq T$, the constrained problem is equivalent to the original unconstrained problem. Further, by Theorem 4.15, the optimal displacement achieved strictly increases as the attack duration T increases, and using $T = M$ achieves the maximum extension distance for any fixed attack size $M \in \mathfrak{N}_0$. It is worth noting that all the results of Section 4.2 remain valid in this constrained domain; We need only rework the results obtained in the last section.

4.4.1 Stacking Blocks of Variable Mass

As was shown in Section 4.3.1, the original problem is equivalent to the problem of optimally extending a stack of identical blocks over the edge of a table—a reduction to a solved problem. Not surprisingly, the time-limited version of the attack on the hypersphere is also analogous to a constrained version of the stacking blocks problem. In this version, we have M points corresponding to M identical blocks. These points must be arranged into T vertical stacks such that all points in a given stack are bound together at the same (horizontal) location, which corresponds to placing points at a single time iteration. Thus, the t^{th} vertical stack contains $\alpha_t \in \mathfrak{N}_0$ blocks of unit weight and has a combined *mass* of α_t. Additionally, to incorporate the initial supporting mass, there is an initial unmovable mass of $\alpha_0 = N$, which rests at the outer edge of the topmost block. The attacker must optimize the grouping of the M blocks such that the resulting T stacks achieve a maximal extension beyond the edge of the table; Figure 4.2 depicts this problem with three stacks. However, this constrained form of the stacking blocks problem is more difficult than the original one since it adds this vertical stacking constraint and, since the size of each stack is integral, this is an integer program.

To the best of our knowledge, there is no generally known (integer) solution for this problem. However, for our purpose of bounding the optimal progress an attacker can

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

achieve, we are not required to find a provably optimal feasible strategy. Instead, if by relaxing the limitations on our adversary (i.e., giving it strictly more power than the problem allows), we can derive an optimal strategy, then the displacement achieved by this optimal relaxed strategy will bound the optimal strategy of the true (limited) adversary. One such intuitive relaxation is to remove the constraint that the vertical stacks contain an integer number of elements; instead, we allow them to be real-valued (but still non-negative). This leads to a new formulation: the attacker has T blocks of equal length but variable mass, and it wants to optimally allocate mass to those blocks so that their total mass is M and they achieve an optimal horizontal displacement beyond the table. This is the continuous variant of the problem: the *variable-mass block-stacking problem*.

By moving into the continuous realm, we consider continuous sequences $\boldsymbol{\beta} \in \mathcal{B}^{(M,T)}$ where $\beta_t \in \Re_{0+}$. For a given T and M, the (relaxed) attacker wants to find an $\boldsymbol{\beta}^\star$ such that for all $\boldsymbol{\beta} \in \mathcal{B}^{(M,T)}$ it achieves $D(\boldsymbol{\beta}^\star) \geq D(\boldsymbol{\beta})$. The optimization problems presented in Equations (4.13) and (4.14) naturally extend to the continuous context, and most of the observations of the original problem carry over to the continuous realm. In particular, it is clear that the location of zero-elements is still irrelevant as was shown in Theorem 4.12, and the zero-elements can again be discarded without affecting optimality. Moreover, the proof of Theorem 4.14 made no use of the fact that α_t were integers; only that $\alpha_t \geq 0$. The same line of reasoning can be applied to $\beta_t \in \Re_{0+}$ and any optimal continuous solution is monotonically increasing. In fact, the only result of Section 4.2 invalidated by this relaxation is Lemma 4.13—in the continuous domain, it is no longer generally optimal to have $\beta_1 = 1$ because any $\beta_1 = \epsilon > 0$ achieves the optimal initial contribution $\delta_1(\boldsymbol{\beta}) = 1$.

4.4.2 An Alternate Formulation

In the continuous mass setting, it is not straightforward to find an optimal strategy $\beta_T^\star = (\beta_t \in \Re_{0+})$ for the program given in Equation (4.14). While the original stacking-blocks problem is a well-known example of a center-of-mass problem with a published solution, we are not aware of research addressing a block-stacking problem in which mass can be redistributed among the blocks. In the sections that follow, we provide a solution to this problem and bound the effort required by the attacker to achieve its goal in the analogous setting.

To solve this problem we return to the intuition given in Remark 4.5 that the attacker must balance current gains against past actions, and we rewrite the problem in terms of the mass accumulated in the t^{th} retraining iteration. In particular, by considering that the relaxed cumulative sum of mass of Equation (4.7) is given by $\mu_t = \sum_{\ell=1}^{t} \beta_\ell$ with $\mu_0 = N$ and $\mu_T = M$, each element of the attack sequence can be rewritten as $\beta_t = \mu_t - \mu_{t-1}$. This allows us to rewrite the entire objective function in terms of the cumulative mass sequence, $\boldsymbol{\mu}$, which results in

$$D(\boldsymbol{\mu}) = T - \sum_{t=1}^{T} \frac{\mu_{t-1}}{\mu_t}, \tag{4.15}$$

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

with $\mu_0 = N$. From the definition of μ_t as a *cumulative mass*, it follows that $\mu_0 \leq \mu_1 \leq \mu_2 \leq \ldots \leq \mu_T = M + N$. Finally, in the T^{th} iteration, the mass must total $M + N$ since, from Theorem 4.15, we have that attacks using less than M total points are nonoptimal, and hence, are excluded from consideration. Thus, optimality can be achieved for the attacker by solving the following program for $\boldsymbol{\mu}^\star = \left(\mu_t^\star\right)$:

$$\boldsymbol{\mu}^\star \in \operatorname{argmax}_{\boldsymbol{\mu}} D(\boldsymbol{\mu}) = T - \sum_{t=1}^{T} \frac{\mu_{t-1}}{\mu_t} \tag{4.16}$$

$$\text{s.t.} \qquad \begin{array}{l} \mu_0^\star \leq \mu_1^\star \leq \ldots \leq \mu_T^\star \\ \mu_0^\star = N, \quad \mu_t^\star \in \Re_+, \quad \mu_T^\star = M + N \end{array}$$

In this reformulation, the total mass constraints still capture every aspect of the relaxed problem, and it is easier to optimize this reformulated version of the problem. This leads to our desired bounds on the optimal progress of an attacker in the time-constrained problem variant.

4.4.3 The Optimal Relaxed Solution

Using the alternative formulation of Program (4.16), we can calculate the optimal relaxed strategy for $T < M$ (for $T \geq M$, Theorem 4.16 applies). The results of this optimization are summarized by the following theorem:

THEOREM 4.18 *For any $N > 0$ and $T < M$, the sequence of masses described by the total mass sequence $\mu_t^\star = N\left(\frac{M+N}{N}\right)^{\left(\frac{t}{T}\right)}$ for $t \in 1 \ldots T$ is the unique solution of Program* (4.16)*. Moreover, this total mass sequence provides the following bound on the optimal displacement alignment*

$$D_N^\star(M, T) \leq D(\boldsymbol{\mu}^\star) = T\left(1 - \left(\frac{N}{M+N}\right)^{1/T}\right). \tag{4.17}$$

Finally, the actual optimal sequence of mass placements $\boldsymbol{\beta}^\star$ can be described by

$$\beta_t^\star = \begin{cases} N, & \text{if } t = 0 \\ N\left(\frac{M+N}{N}\right)^{\frac{t-1}{T}}\left(\left(\frac{M+N}{N}\right)^{\frac{1}{T}} - 1\right), & \text{if } t \in 1 \ldots T \end{cases}. \tag{4.18}$$

Also note that, as required this solution meets the conditions $\mu_0 = N$, $\sum_{t=1}^{T} \beta_t^\star = \mu_T = M + N$, and for all $t > 0$, $\mu_{t-1} \leq \mu_t$.

Proof. See Appendix B.5. □

In general, the optimal relaxed strategy of Equation (4.18) does not produce integer strategies except in the case when $\left(\frac{M+N}{N}\right)^{\frac{1}{T}} \in \mathfrak{N}$. Thus, these strategies are not generally optimal according to the program given in Equation (4.14). Moreover, it is nontrivial to convert an optimal relaxed strategy to an optimal integer-valued one (rounding can produce good strategies but is not necessarily optimal). However, we need not explicitly compute the optimal integer-valued strategy to quantify its impact.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

The utility of this result is that it allows us to bound the optimal displacement achieved by the optimal integer-valued attack sequence and subsequently invert these bounds using Lemma 4.1 since the function $T\left(1-\left(\frac{N}{M+N}\right)^{1/T}\right)$ is monotonically increasing in both M and T. Also, in agreement with Theorem 4.6, this function is upper bounded by T and has an upper limit (as $T \to \infty$) of $\log\left(\frac{M+N}{N}\right)$. For any fixed T and M, the displacement achieved is at most $\min\left[T, \log\left(\frac{M+N}{N}\right)\right]$. The result is as follows:

$$M^\star \geq \begin{cases} N\left(\frac{T}{T-D_R}\right)^T - N \quad \geq \quad N\left(\exp\left(D_R\right)-1\right), & \text{if } D_R < T \\ \infty, & \text{if } D_R \geq T \end{cases}, \tag{4.19}$$

where the second case of this bound reflects the restriction that the total relative displacement cannot exceed the attack duration T, regardless of how many attack points are used (see Theorem 4.6). Similarly, the minimum number of retraining iterations required to achieve the displacement D_R for a given $N, M > 0$ can be determined as solutions to the following inequality

$$\left(\frac{D_R}{T}-1\right)^T \geq \frac{N}{M+N},$$

which is computable using the Lambert-W function (i.e., the inverse of the function $f(w) = w\exp(w)$), but cannot be expressed in terms of elementary functions and does not contribute to our intuition about the problem except to say that the bound can be computed (see Figure 4.3).

We have now provided strong bounds on the effort required of the adversary to achieve its desired goal. However, before we conclude this section, note that the result

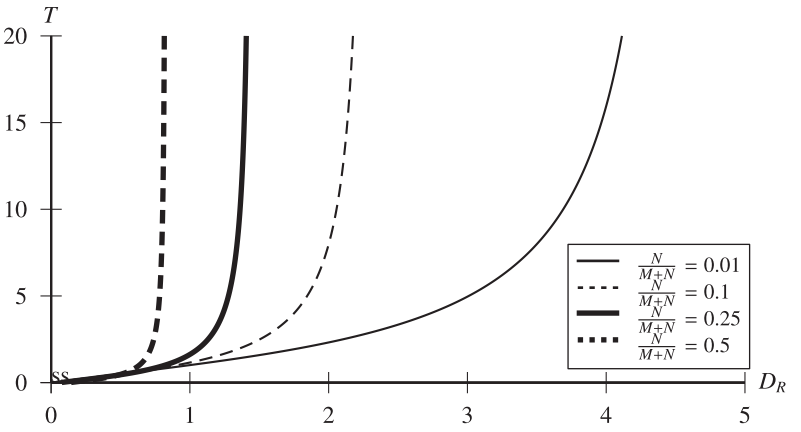


Figure 4.3 Plots depicting the lower bound on the number of retraining iterations T required to make the goal displacement D_R attainable. Each curve shows the lower bound for a particular fixed ratio $\frac{N}{M+N}$. As one expects, when $M \gg N$ the requirement is lessened, but in each case, there is a turning point in the curve where the bound sharply increases; e.g., it is practically impossible to achieve $D_R > 1$ when $\frac{N}{M+N} = 0.5$ because it would require an unreasonable attack duration to do so. The reader should note that, additionally, these lower bounds are loose unless $M \gg T$.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

in Equation (4.19) is only applicable when $N \geq 1$. Briefly, we address this special case and then consider additional scenarios for attacks against hypersphere detectors.

4.4.3.1 Attacks against a Nonsupported Initial Hypersphere

As noted earlier, unlike Theorem 4.16, Theorem 4.18 and the subsequent bounds that result from it do not apply when $N = 0$. This is because, without an initial constraint on the sequence, increasingly large displacements can be obtained by starting with an ever-diminishing initial point, $\mu_1 > 0$. The problem is that for $N = 0$, the initial hypersphere centered at $\mathbf{c}^{(0)}$ is assumed to have no initial mass. Thus, the first mass placed by the attacker on the boundary displaces the mean to $\mathbf{c}^{(1)} = R$ regardless of its size. In the integer-valued case, this assumption has little effect on the outcome due to Lemma 4.13, but in the continuous case, this leads to an attack that places minuscule (but exponentially increasing) masses in the initial phase of the attack and then adds the overwhelming majority of the total mass in the final stages.

In the continuous domain, we can examine the sequence of optimal attacks given by Theorem 4.18 in the limit as $N \to 0$. In doing so, we derive that the optimal distance achieved by Equation (4.17) approaches T; i.e., $\lim_{N\to 0} D_N^\star(M, T) = T$. As shown in Theorem 4.6, this is, if fact, the maximal possible displacement alignment attainable by any sequence of T duration. However, such attack sequences do not correspond well to the feasible integer-valued attacks and do not improve our bounds.

To provide better bounds for the case of $N = 0$, we reintroduce the constraint $\mu_1 = \beta_1 = 1$ from the original integer-valued problem. That is, we now assume that the result of Lemma 4.13 hold; this constrains the continuous-valued sequences to more closely match the integer-valued ones. This is a reasonable assumption to make since the attack does not begin until the attacker uses at least one attack point as discussed in Remark 4.9.

This new constraint for the problem leads to a similar problem as was analyzed earlier in Equation (4.16) using $\mu_1 = 1$ instead of $\mu_0 = N$, and the subsequent results mirror those presented in Theorem 4.18 and its proof. In particular, we have that the total mass sequence given by $\mu_t^\star = M^{\frac{t-1}{T-1}}$ for $t \in 1 \ldots T$ is the unique solution and achieves an optimal displacement alignment of $D_0^\star(M, T) \leq T - (T-1) \cdot M^{\frac{-1}{T-1}}$. Again, this bounding function is monotonically increasing in both M and T, which leads to the following bound on the adversary's effort when $N = 0$:

$$M^\star \geq \begin{cases} \left(\frac{T}{T-D_R}\right)^T & \geq \quad \exp(D_R - 1), \quad \text{if } D_R < T \\ \infty, & \qquad\qquad\qquad\quad \text{if } D_R \geq T \end{cases},$$

where the second case again reflects the restriction that the total relative displacement cannot exceed the attack duration T. Also, a bound on the minimum number of retraining iterations, $T^\star$, required to achieve the displacement D_R for a given $M > 0$ can be computed from the above bound on $D_0^\star(M, T)$, but it cannot be expressed in terms of elementary functions and does not contribute further insight.

This concludes our results for attacks against the hypersphere model described in Section 4.1. In all cases we have thus far examined, the impact of the attacker on the

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

model was extremely limited, and the number of attack points required to attain a desired displacement D_R was minimally exponential in D_R. These results provide a strong guarantee on the hypersphere's security, but, as discussed in the next section, the retraining model used is overly rigid. We now proceed by examining alternative retraining models.

4.5 Attacks against Retraining with Data Replacement

Now we consider an alternative learning scenario, in which new data replaces old data, allowing the hypersphere detector to adapt more agilely. This scenario was explored by Kloft & Laskov (2012), and here we summarize their results. We assume that each new point is introduced by the attacker and *replaces* exactly one existing point that was previously used to train the hypersphere in the last retraining iteration. This alters the centroid update formula given in Equation (4.6) to

$$\mathbf{c}^{(t)} = \mathbf{c}^{(t-1)} + \frac{1}{N}\left(\mathbf{a}^{(t)} - \mathbf{x}_{rep}^{(t)}\right), \tag{4.20}$$

where $\mathbf{x}_{rep}^{(t)}$ is the point to be replaced by $\mathbf{a}^{(t)}$. Notice that, unlike in Section 4.2, the mass supporting the new hypersphere's centroid does not change since we have both added and removed a point. As we show below, in this scenario, the attacker is no longer inhibited by past attack points, which makes its attack *considerably* more effective than in previous attack scenarios.

One can generalize this setting to again allow the attacker to use α_t attack points in each iteration (generally with $\alpha_t \in \{0, \ldots, N\}$). However, doing so considerably complicates the subsequent analysis both in terms of choosing the set of optimal attack vectors $\mathbb{A}^{(t)}$ and optimally apportioning the attack points into an overall strategy $\boldsymbol{\alpha}$. Moreover, as we saw in Section 4.3, the strategy of placing a single attack point at each iteration ($\alpha_t = 1$) is the optimal strategy for placing M points without any time constraints and strictly dominates all time-constrained strategies. Hence, in this section, to simplify our presentation, we focus only on single-point attack strategies (assuming that $T = M$) and comment on the effects of this assumption.

Under this single-point replacement scenario, the *relative displacement* and *total relative displacement* are given, respectively, by

$$\mathbf{r}_t = \tfrac{1}{R \cdot N}\left(\mathbf{a}^{(t)} - \mathbf{x}_{rep}^{(t)}\right) \quad \text{and} \quad \mathbf{D}_T = \tfrac{1}{R \cdot N}\sum_{t=1}^{T}\left(\mathbf{a}^{(t)} - \mathbf{x}_{rep}^{(t)}\right).$$

Unlike the results derived in Section 4.2 and their subsequent consequences, we require more information about the specific replacement policy used by the hypersphere to optimize or analyze the impact of attacks in this scenario. Next we discuss various replacement policies for choosing $\mathbf{x}_{rep}^{(t)}$ and their effect on the attack's success. However, from the expression of $\mathbf{D}_T$ given above, it is obvious that the attacks will generally be more successful than those analyzed in previous sections. In fact, note that if for all t we have $\left(\mathbf{a}^{(t)} - \mathbf{x}_{rep}^{(t)}\right)^{\top}\left(\mathbf{x}^{A} - \mathbf{c}^{(0)}\right) \geq \kappa$ for some fixed constant $\kappa > 0$, then the attacker

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

can achieve a displacement alignment of at least

$$\rho\left(\mathbf{D}_T\right) \geq \frac{\kappa}{RN\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} T.$$

This suggests that, under replacement, attacks may potentially achieve a linear displacement alignment that *linearly increases* with the attack duration T using only a *single* attack point at each iteration (hence a total of $M = T$ attack points). This would be an astounding success for the attacker, especially compared to the exponential results that were demonstrated for retraining without replacement.

Below we discuss a number of potential replacement policies and how they affact the attacker's success. In this discussion, we will consider policies and effects that are random. To do so, we analyze attacks that are optimal for each step of the attack, but are not necessarily optimal with respect to the entire attack strategy. For this purpose, we consider the following notion of greedy optimal attacks.

DEFINITION 4.19 At the t^{th} iteration of the attack, given the current centroid $\mathbf{c}^{(t-1)}$, an attack using attack point $\mathbf{a}^{(t)}$ is a *greedy optimal attack* if it optimizes

$$\mathrm{E}\left[\rho\left(\mathbf{D}_t\right) \;\middle|\; \mathbf{c}^{(t-1)}\right] \tag{4.21}$$

subject to the constraint that $\left\|\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right\| \leq R$.

4.5.1 Average-out and Random-out Replacement Policy

First, we examine two simple replacement policies: removing a copy of the previous centroid (*average-out replacement*) and removing a random point from the data (*random-out replacement*). These policies have a predictable impact on the displacement alignment, which allows the attacker to achieve its objective using relatively few attack points.

In *average-out replacement*, the point that is replaced by any new data point is always a copy of the current centroid; i.e., in the t^{th} iteration, $\mathbf{x}_{rep}^{(t)} = \mathbf{c}^{(t-1)}$. Thus, from Equation (4.20), the t^{th} centroid is given by $\mathbf{c}^{(t)} = \mathbf{c}^{(t-1)} + \frac{1}{N}\left(\mathbf{a}^{(t,\ell)} - \mathbf{c}^{(t-1)}\right)$, which yields a result similar to Theorem 4.7; namely, the optimal attack point at every iteration is $\mathbf{a}^{(t)} = \mathbf{c}^{(t-1)} + R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$ and the optimal T^{th} centroid is $\mathbf{c}^{(T)} = \mathbf{c}^{(0)} + \frac{RT}{N} \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$. This yields the following optimal attack parameters:

$$\mathbf{r}_t = \frac{1}{N} \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \quad \forall\, t \in \{1, \ldots, T\}$$

$$\mathbf{D}_T = \frac{T}{N} \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}.$$

Further, the resulting displacement alignment is thus $\rho\left(\mathbf{D}_T\right) = \frac{T}{N}$. In accordance with our discussion above, the relative displacement achieved under this policy at each iteration achieves a fixed inner product with the desired direction $\mathbf{x}^A - \mathbf{c}^{(0)}$ of $\kappa = \frac{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}{N}$. Each attack point contributes $\frac{1}{N}$ of a unit step in the desired direction, and the desired

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

displacement can be achieved with $M^\star = T^\star = N \cdot D_R$; i.e., the goal only requires linearly many points in the desired relative displacement D_R.

Remark 4.20 Above, the attacker can optimally displace the hypersphere by using one attack point per iteration. However, note that the impact achieved on the centroid by each attack point is the same regardless of how many attack points are used in any iteration. The per-point impact is the same regardless of how many points are used in each iteration. In fact, if N attack points are used in each iteration, the displacement alignment is $\rho(\mathbf{D}_T) = T$, the maximum possible displacement alignment when replacement was not permitted (see Theorem 4.6). The attacker's goal can be achieved using only $T^\star = D_R$ iterations, although the number of points required remains as $M^\star = N \cdot D_R$. As such, it is more appropriate to write $\rho(\mathbf{D}_T) = M$, which holds for average-out replacement regardless of the allocation strategy.

For the *random-out replacement* policy, $\mathbf{x}_{rep}^{(t)}$ is a randomly selected element of the hypersphere's current training set. As such, it is no longer possible to compute the attack parameters precisely—namely, the terms $\left(\mathbf{a}^{(t)} - \mathbf{x}_{rep}^{(t)}\right)$ that are used to recursively compute the hypersphere's centroid depend on a random variable. However, we can consider greedy optimal attacks that locally optimize the *expected* displacement alignment at each iteration t with respect to the centroid $\mathbf{c}^{(t-1)}$ obtained from the previous iteration. In particular, the expected value of $\rho(\cdot)$ can be simplified by noting that $\mathrm{E}\left[\mathbf{D}_t \;\middle|\; \mathbf{c}^{(t-1)}\right] = \frac{\mathrm{E}\left[\mathbf{c}^{(t)} \;\middle|\; \mathbf{c}^{(t-1)}\right] - \mathbf{c}^{(0)}}{R}$. By the linearity of expectations and the definition of displacement alignment in Equation (4.5) we have

$$\mathrm{E}\left[\rho(\mathbf{D}_t) \;\middle|\; \mathbf{c}^{(t-1)}\right] = \frac{\mathrm{E}\left[\mathbf{c}^{(t)} \;\middle|\; \mathbf{c}^{(t-1)}\right]^\top \left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)}{R\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} - \frac{\left(\mathbf{c}^{(0)}\right)^\top \left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)}{R\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|},$$

where the second term is a constant determined by the parameters of the problem. Thus, we seek to maximize the numerator of the first term, for which $\mathrm{E}\left[\mathbf{c}^{(t)} \;\middle|\; \mathbf{c}^{(t-1)}\right] = \mathbf{c}^{(t-1)} + \frac{1}{N}\left(\mathbf{a}^{(t)} - \mathrm{E}\left[\mathbf{x}_{rep}^{(t)} \;\middle|\; \mathbf{c}^{(t-1)}\right]\right)$ since $\mathbf{a}^{(t)}$ is not considered a random variable.

In general, the attacker does not know the distribution of the candidate replacement data points $\left\{\mathbf{x}^{(\ell)}\right\}$, which consist of a mixture of benign and adversarial points. However, it does know that they have an empirical mean of $\mathbf{c}^{(t-1)}$, since these data points are the sample used to center the hypersphere. Since the replacement point is selected randomly from this set (with equal probability), the required expectation is $\mathrm{E}\left[\mathbf{x}_{rep}^{(t)} \;\middle|\; \mathbf{c}^{(t-1)}\right] = \mathbf{c}^{(t-1)}$. Thus, as with average-out replacement, the optimal greedy attack point is $\mathbf{a}^{(t)} = \mathbf{c}^{(t-1)} + R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$ and the expected displacement achieved is

$$\mathrm{E}\left[\mathbf{r}_t \;\middle|\; \mathbf{c}^{(t-1)}\right] = \frac{1}{N} \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \quad \forall\, t \in \{1, \ldots, T\}$$

$$\mathrm{E}\left[\mathbf{D}_T \;\middle|\; \mathbf{c}^{(t-1)}\right] = \sum_{t=1}^{T} \mathrm{E}\left[\mathbf{r}_t \;\middle|\; \mathbf{c}^{(t-1)}\right] = \frac{T}{N} \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}.$$

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

Naturally, the expected displacement alignment is again $\mathrm{E}\left[\rho\left(\mathbf{D}_T\right) \;\middle|\; \mathbf{c}^{(t-1)}\right] = \frac{T}{N}$. Thus, randomly selecting the point to be replaced does not deter the attacker's expected progress compared to the average-out policy.

4.5.2 Nearest-out Replacement Policy

Here we consider a replacement rule that is intended to diminish the success of poisoning when old data is replaced by new data. In particular, we consider *nearest-out replacement*, in which each new datum replaces the old data point that lies closest to it. This policy is designed to reduce the effectiveness of attacks because it limits the total displacement caused by any attack point. However, under the assumption that the adversary knows all training data, Kloft & Laskov (2012) showed that an adversary can use a greedy optimization procedure to find the optimal point to insert conditioned on the current training data—in this case, such a strategy is *greedy* since it does not factor future gains when selecting the next best point to insert.

To counter nearest-out replacement, the strategy employed by the adversary is to find the best point to replace the j^{th} point in the dataset; i.e., the point $\mathbf{a}^{(t,j)}$ that (i) lies within the t^{th} hypersphere, (ii) will replace $\mathbf{x}^{(j)}$, and (iii) has the largest displacement alignment of any such point . To find this point, consider that the N data points divide $\mathcal{X}$ into N regions called *Voronoi cells*; the j^{th} Voronoi cell is the set of points that are closer to $\mathbf{x}^{(j)}$ than any other data point in the dataset. As such, the sought-after point $\mathbf{a}^{(t,j)}$ must lie within the j^{th} Voronoi region and thus can be found by solving the following optimization problem:

$$\begin{aligned} \mathbf{a}^{(t,j)} = \operatorname{argmax}_{\mathbf{x}} \; & \tfrac{1}{RN}\left(\mathbf{x} - \mathbf{x}^{(j)}\right)^{\top} \frac{\left(\mathbf{x}^{A} - \mathbf{c}^{(t)}\right)}{\left\|\mathbf{x}^{A} - \mathbf{c}^{(t)}\right\|} \\ \text{s.t.} \quad & \forall\, k \in 1, \ldots, N \quad \left\|\mathbf{x} - \mathbf{x}^{(j)}\right\| \leq \left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| \\ & \left\|\mathbf{x} - \mathbf{c}^{(t)}\right\| \leq R. \end{aligned} \tag{4.22}$$

The objective of this program maximizes the displacement alignment for replacing the j^{th} point, the first constraint requires that the new point lie within the j^{th} Voronoi cell, and the second constraint requires it to lie within the t^{th} hypersphere. The attacker can thus solve for the best points relative to each of the N data points and select the one that achieves the largest displacement alignment as the t^{th} attack point, $\mathbf{a}^{(t)}$, as depicted in Figure 4.4. This process is repeated at each attack iteration.

The program of Equation (4.22) is a quadratically constrained linear program, for which the quadratic constraints on $\mathbf{a}^{(t,j)}$ can be expressed in terms of a positive definite matrix. Thus, the programs are convex and have a unique optimum (Boyd & Vandenberghe 2004). They can generally be solved by convex optimizers, but current solvers do not scale well when N is large. However, an alternative approach is presented in Algorithm 4.1, which utilizes a quadratic program instead. This optimization problem minimizes the radius of the point within the neighborhood of the k^{th} data point, but is constrained to obtain a minimum displacement alignment, $\hat{\rho}$. If such a point is feasible and lies within the radius R of $\mathbf{c}^{(t)}$, then $\hat{\rho}$ is a lower bound on the displacement

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

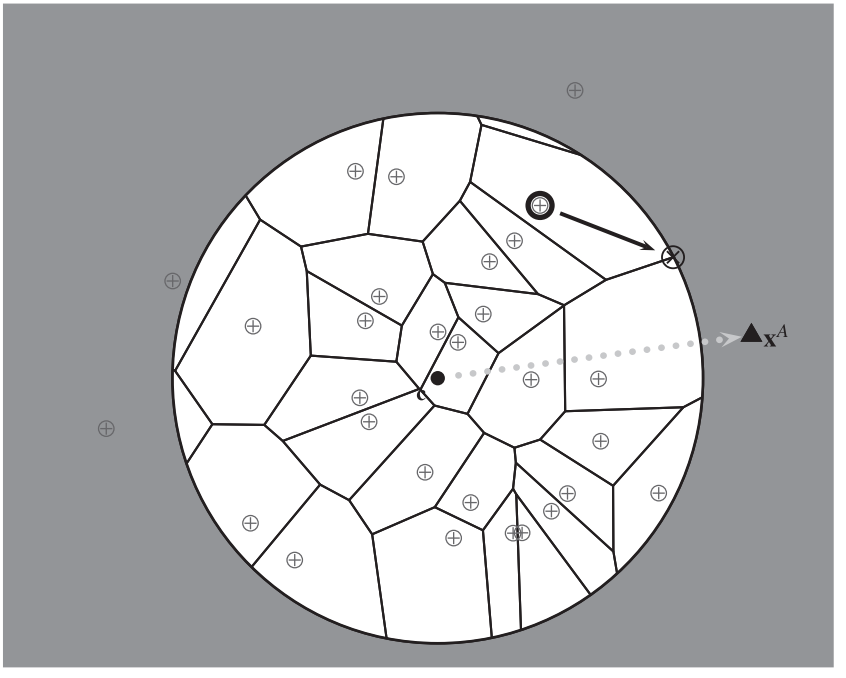


Figure 4.4 A depiction of an iteration of the optimal greedy attack against a hypersphere retrained using nearest-out replacement. The attacker wants to shift the current centroid $\mathbf{c}$ toward the target point $\mathbf{x}^A$, and the desired displacement direction is shown with a gray vector between them. Each training point is indicated with a $\oplus$. These points induce a Voronoi decomposition of the space depicted by the black grid within the hypersphere. Each such Voronoi cell is the set of points that would replace the enclosed training point. Finally, the optimal attack point is represented by a $\otimes$—it replaces the indicated training point and thereby yields the maximum possible displacement alignment according to Program (4.22).

alignment that can be achieved by replacing the k^{th} point; otherwise, it is an upper bound. Thus, we can perform a binary search for the maximum attainable displacement alignment that can be achieved relative to each point. Further, since we are searching for the maximum possible displacement alignment, we can initialize the initial lower bound of the k^{th} point to the maximum displacement alignment thus far achieved for the previous $(k-1)$ points. This overall procedure is captured in Algorithm 4.1.

However, there remains one aspect of this problem we have not yet addressed. Until now, we implicitly assumed that the Voronoi region of each point has a non-empty intersection with the hypersphere, but this assumption may be violated after many iterations of greedy optimal attacks. Such points are *abandoned* and act as a drag on the attack since they are no longer replaceable and lie far from the desired target $\mathbf{x}^A$. However, the attacker can prevent points from being abandoned by finding the optimal attack point, determining (through simulation) if the attack would cause any points to be abandoned, and, if so, finding the optimal attack point for the abandoned point. By ensuring that no points are abandoned, the attacker loses gain in that iteration, but prevents a long-term

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

ALGORITHM 4.1 NEAREST-OUT GREEDY ATTACK

$Nout - Opt\left(\mathbf{x}^A, \mathbf{c}^{(0)}, R, \mathbf{c}^{(t)}, \left\{\mathbf{x}^{(j)}\right\}, \epsilon\right)$
Let $\rho^- \leftarrow -2 \cdot R$
for all $j \in 1, \ldots, N$ **do begin**
  Let $\rho^+ \leftarrow 2 \cdot R$
  while $\rho^+ - \rho^- > \epsilon$ **do begin**
    Solve for

$$\mathbf{a}^{(t,j)} = \text{argmin}_{\mathbf{x}} \left\|\mathbf{x} - \mathbf{c}^{(t)}\right\| \tag{4.23}$$
$$\text{s.t.} \quad \begin{array}{l} \forall\, k \in 1, \ldots, N \quad 2\left(\mathbf{x}^{(k)} - \mathbf{x}^{(j)}\right)^\top \mathbf{x} \leq \left\|\mathbf{x}^{(k)}\right\|^2 - \left\|\mathbf{x}^{(j)}\right\|^2 \\ \frac{1}{RN}\left(\mathbf{x} - \mathbf{x}^{(j)}\right)^\top \frac{\left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \geq \frac{\rho^+ - \rho^-}{2} \end{array}$$

    if Program 4.23 is feasible **and** $\left\|\mathbf{a}^{(t,j)} - \mathbf{c}^{(t)}\right\| \leq R$ **then** $\rho^- \leftarrow \frac{\rho^+ - \rho^-}{2}$ **and** $\mathbf{a}^{(t)} \leftarrow \mathbf{a}^{(t,j)}$
    else $\rho^+ \leftarrow \frac{\rho^+ - \rho^-}{2}$
  end while
end for
return: $\mathbf{a}^{(t)}$

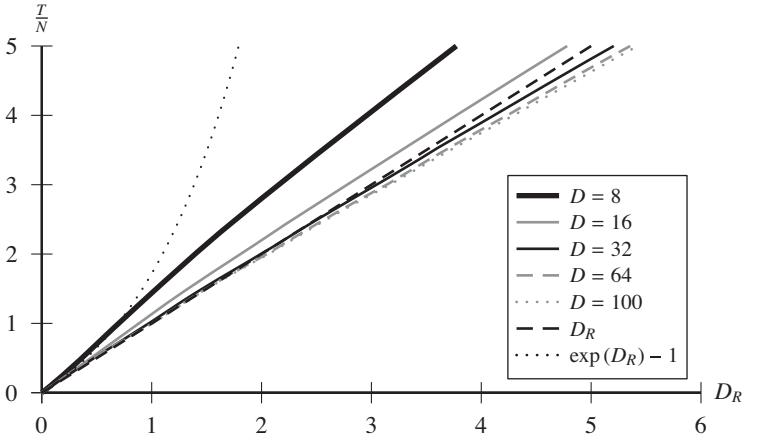
drag on its attack. This makes the attack more globally optimal, but more difficult to analyze precisely.

Because of this problem, there is no known exact result for this attack's total displacement alignment over T attack iterations. However, we can approximate it. Namely, in the worst case for the attacker, all training points would be co-linear along the direction $\mathbf{x}^A - \mathbf{c}^{(t)}$. As such, we can analyze the one-dimensional case. Here, assuming that no points will be abandoned, the displacement achieved by a single attack is at least $\frac{R}{2N}$ since, at worst, the N points are spread evenly between the centroid and the radius R. Thus, the total displacement is at least $\frac{1}{2N^2}$ times the number of iterations in which no points are abandoned. However, in practice, the gains are much greater in high-dimensional problems and are approximately linear in $\frac{T}{N}$. To see this, we followed the experimental procedure of Kloft & Laskov (2012) using $N = 100$ initial data points drawn from a standard normal distribution in $D \in \{2, 4, 8, 16, 32, 64, 100\}$ dimensions with a radius R selected to have a false-negative rate of 0.001. From this initial setting, we constructed over $T = 5 \cdot N = 500$ iterations of greedy attacks. The experiments were repeated 10 times, and the results are shown in Figure 4.5. As can be seen in these plots, the effects are approximately linear for $D > 4$ with a slope that approaches and even can perform slightly better than $D_R = \frac{T}{N}$.

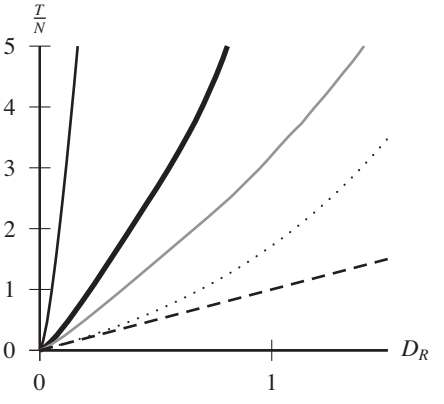
4.6 Constrained Attackers

Reiterating our results thus far, in Sections 4.3 and 4.4, we showed that a hypersphere detector, which uses bootstrap retraining without any data replacement, is resilient to

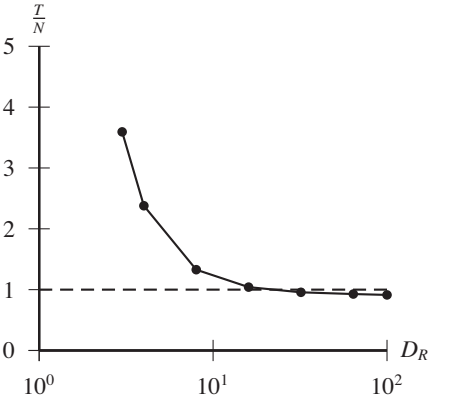
https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press



(a) Greedy Attack on Nearest-out Policy: High Dimensional



(b) Greedy Attack on Nearest-out Policy: Low Dimensional



(c) Slope as a Function of Dimension

Figure 4.5 These plots show the empirical effect of iterative greedy attacks against a hypersphere using nearest-out replacement. **(a)** In high dimensions, the required duration of the attack increases approximately linearly as a function of D_R, with a slope that decreases as dimension increases. **(b)** In small dimensions, the required duration can exceed the exponential bound due to the dense clustering of the data in the hypersphere. **(c)** When approximated as a linear function, the slope of the fit line decreases as the dimension of the hypersphere increases. For $D \approx N$, the slope can be slightly less than one.

attacks in the sense that an attacker must use exponentially many attack points in terms of its desired displacement, D_R. However, without any data replacement, the hypersphere becomes unadaptable as more data is received and retraining quickly becomes futile. However, as we saw in the last section, when data replacement is incorporated, an attacker can achieve its desired displacement, D_R, of the hypersphere detector using only linearly many attack points to do so under several possible replacement policies. These results suggest that having an adaptive hypersphere detector may be incompatible

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

with having a model that is difficult for an attacker to coerce; i.e., iterative relearning and security are not simultaneously possible for hypersphere learning. However, to this point, we have been exceedingly pessimistic in assuming that the attacker can control all data points once the attack commences. In this section, we examine more realistic assumptions on the attacker's capabilities and show that in some settings, iteratively retrained hyperspheres are more resilient to poisoning attacks than indicated by the previous worst-case analysis. As with the last section, here we summarize results presented by Kloft & Laskov (2012).

To simplify the analysis, here we only consider *average-out replacement*, and we restrict ourselves to the scenario in which the hypersphere is retrained whenever it receives a new point; i.e., $\alpha_t = 1$ for all t. Through this restriction, we need only consider how the attacker designs a single optimal point $\mathbf{a}^{(t)}$ at the t^{th} iteration, and we assume it does so greedily (i.e., only considering the current hypersphere).

In contrast to previous sections, we now assume that there are two sources of new data: attack data generated by the attacker and benign data generated by other users of the system. We assume that the benign data (i) comes from a natural distribution $P_{\mathbf{x}}$ that is neither advantageous nor detrimental to the adversary, (ii) is drawn independently and identically from that distribution, (iii) is randomly interleaved with the adversarial data, and most importantly, (iv) is always accepted for retraining regardless of the current state of the classifier (i.e., bootstrap retraining is relaxed for benign data).[2] In particular, we assume that each new data point given to the classifier is randomly selected to be either adversarial or benign according to a Bernoulli random variable with fixed parameter $\nu \in [0, 1]$; i.e., when the t^{th} new data point is introduced, it is either the point $\mathbf{a}^{(t)}$ selected by the adversary with probability ν or it is a point $\mathbf{x}^{(t)} \sim P_{\mathbf{x}}$ with probability $1 - \nu$ and the attacker cannot alter the probability ν of its point being selected. Equivalently, we can model the new point $\mathbf{x}_{new}^{(t)}$ with a random variable $B^{(t)} \sim \text{Bern}(\nu)$ such that

$$\mathbf{x}_{new}^{(t)} = B^{(t)}\mathbf{a}^{(t)} + \left(1 - B^{(t)}\right)\mathbf{x}^{(t)} \tag{4.24}$$

where $\mathbf{x}^{(t)} \sim P_{\mathbf{x}}$ and $B^{(t)} \in \{0, 1\}$. Importantly, in selecting $\mathbf{a}^{(t)}$, we assume the adversary does not know $B^{(t)}$ or $\mathbf{x}^{(t)}$ but can still observe $\mathbf{c}^{(t)}$ that results and thus can compute $\mathbf{x}_{new}^{(t)}$ after retraining occurs. As before, the adversary must also choose $\mathbf{a}^{(t)}$ to be accepted for retraining, but here we assume that when $B^{(t)} = 0$, the benign $\mathbf{x}_{new}^{(t)}$ will always be accepted. Next, we discuss how the attacker can select $\mathbf{a}^{(t)}$ and analyze its impact under several constraints.

4.6.1 Greedy Optimal Attacks

In the scenario discussed above, new training data is a mixture of attack and benign data, but the attacker *cannot* alter the mixing ratio between them. Under this setting, we assume that the attacker produces an attack point at each iteration to optimize the expected displacement alignment according to Equation (4.21), and either this point

[2] This assumption is removed in alternative models studied by Kloft & Laskov (2012) that are not discussed here.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

or a benign point $\mathbf{x}^{(t)} \sim P_{\mathbf{x}}$ will be used in retraining the hypersphere to obtain the t^{th} centroid. Under average-out replacement (see Section 4.5.1), the outcome of this attack can be described by the resulting centroid and displacement vector, which are computed from $\mathbf{x}_{new}^{(t)}$ as

$$\begin{aligned}
\mathbf{c}^{(t)} &= \mathbf{c}^{(t-1)} + \frac{1}{N}\left(\mathbf{x}_{new}^{(t)} - \mathbf{c}^{(t-1)}\right) \\
&= \mathbf{c}^{(t-1)} + \frac{1}{N}\left(B^{(t)}\left(\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right) + \left(1 - B^{(t)}\right)\left(\mathbf{x}^{(t)} - \mathbf{c}^{(t-1)}\right)\right) \\
\mathbf{D}_T &= \frac{1}{R \cdot N}\sum_{t=1}^{T}\left(\mathbf{x}_{new}^{(t)} - \mathbf{c}^{(t-1)}\right) \\
&= \frac{1}{R \cdot N}\sum_{t=1}^{T}\left(B^{(t)}\left(\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right) + \left(1 - B^{(t)}\right)\left(\mathbf{x}^{(t)} - \mathbf{c}^{(t-1)}\right)\right)
\end{aligned}$$

Due to the structure of these recursive expressions, it is difficult to optimize $\mathbf{D}_T$ over the entire attack sequence. However, given the centroid $\mathbf{c}^{(t-1)}$ from the last iteration, we can derive greedy optimal actions for the adversary under the assumption that all points $\left\{\mathbf{x}^{(t)}\right\}$ are drawn independently from the distribution $P_{\mathbf{x}}$. The result is given by the following lemma.

LEMMA 4.21 *Under average-out replacement, at the t^{th} attack iteration, the greedy optimal attack point is given by*

$$\mathbf{a}^{(t)} = \mathbf{c}^{(t-1)} + R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}.$$

Proof. From Equation (4.21), the greedy optimal strategy optimizes $\mathrm{E}\left[\rho\left(\mathbf{D}_t\right) \mid \mathbf{c}^{(t-1)}\right]$ but since $\mathrm{E}\left[\rho\left(\mathbf{D}_{t-1}\right) \mid \mathbf{c}^{(t-1)}\right] = \rho\left(\mathbf{D}_{t-1}\right)$—a fixed quantity relative to the attacker's actions in the t^{th} step—the former is equivalent to optimizing $\mathrm{E}\left[\rho\left(\mathbf{D}_t\right) - \rho\left(\mathbf{D}_{t-1}\right) \mid \mathbf{c}^{(t-1)}\right]$; i.e., to optimizing the dot product of $\mathbf{r}_t$ with the desired direction $\mathbf{x}^A - \mathbf{c}^{(0)}$. This relative displacement is given by $\mathbf{r}_t = \frac{B^{(t)}}{R \cdot N} \cdot \left(\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right) + \frac{\left(1 - B^{(t)}\right)}{R \cdot N} \cdot \left(\mathbf{x}^{(t)} - \mathbf{c}^{(t-1)}\right)$. Computing the required expected value thus becomes

$$\begin{aligned}
\mathrm{E}\left[\rho\left(\mathbf{D}_t\right) - \rho\left(\mathbf{D}_{t-1}\right) \mid \mathbf{c}^{(t-1)}\right] &= \mathrm{E}\left[\mathbf{r}_t \mid \mathbf{c}^{(t-1)}\right]^{\top} \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \\
&= \frac{\nu}{R \cdot N}\left(\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right)^{\top} \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} \\
&\quad + \frac{1 - \nu}{R \cdot N}\left(\mathrm{E}\left[\mathbf{x}^{(t)} \mid \mathbf{c}^{(t-1)}\right] - \mathbf{c}^{(t-1)}\right)^{\top} \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|},
\end{aligned}$$

where $\mathrm{E}\left[\mathbf{x}^{(t)} \mid \mathbf{c}^{(t-1)}\right]$ is a fixed quantity since $\mathbf{x}^{(t)}$ is drawn independently from $P_{\mathbf{x}}$. By linearity, maximizing this quantity is equivalent to maximizing $\left(\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right)^{\top} \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}$ with respect to $\left\|\mathbf{a}^{(t)} - \mathbf{c}^{(t-1)}\right\| \leq R$. As we saw in Section 4.5.1, this yields the claimed form for optimal $\mathbf{a}^{(t)}$. □

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

4.6.2 Attacks with Mixed Data

Here we analyze the expected net effect of applying the optimal greedy attack of Lemma 4.21 over T iterations and thus provide an analysis for the mixed data scenario described in Equation (4.24). To do so, we require an additional assumption about the benign data's distribution—namely we assume (i) that all benign data is drawn independently from $P_{\mathbf{x}}$, (ii) that the benign data has a stationary mean, $\mathrm{E}_{\mathbf{x}\sim P_{\mathbf{x}}}[\mathbf{x}] = \mathbf{c}^{(0)}$, and (iii) that *benign data is never rejected*. These are strong assumptions about the benign data, but, in assuming that the benign data has a stationary mean and is always accepted, these are conservative assumptions on the attacker and yield the following theorem.[3]

THEOREM 4.22 (Paraphrased from Kloft & Laskov 2012) *Given a fixed mixture probability ν, applying the greedy optimal attack strategy (given by Lemma 4.21) at each iteration yields an expected displacement alignment of*

$$\mathrm{E}\left[\rho\left(\mathbf{D}_T\right)\right] = \frac{\nu}{1-\nu} \cdot \left(1 - \left(1 - \frac{(1-\nu)}{N}\right)^T\right) \leq \frac{\nu}{1-\nu}$$

after T iterations.

Proof. Under the optimal attack strategy of Lemma 4.21 the centroid becomes

$$\begin{aligned}
\mathbf{c}^{(t)} &= \mathbf{c}^{(t-1)} + \frac{1}{N}\left(B^{(t)} R \cdot \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} + \left(1 - B^{(t)}\right)\left(\mathbf{x}^{(t)} - \mathbf{c}^{(t-1)}\right)\right) \\
&= \mathbf{c}^{(t-1)} + \frac{B^{(t)} R}{N} \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|} + \frac{1}{N}\left(1 - B^{(t)}\right)\left(\mathbf{x}^{(t)} - \mathbf{c}^{(0)}\right) \\
&\quad - \frac{1}{N}\left(1 - B^{(t)}\right)\left(\mathbf{c}^{(t-1)} - \mathbf{c}^{(0)}\right),
\end{aligned}$$

where the summation has been conveniently reorganized for later. Now, using the definition of $\mathbf{D}_t = \frac{\mathbf{c}^{(t)} - \mathbf{c}^{(0)}}{R}$ from Equation (4.4), we substitute this form of $\mathbf{c}^{(t)}$ and use the linearity of $\rho(\cdot)$ to obtain

$$\begin{aligned}
\mathbf{D}_t &= \left(1 - \frac{1 - B^{(t)}}{N}\right)\mathbf{D}_{t-1} + \frac{\mathbf{x}^A - \mathbf{c}^{(0)}}{\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}\frac{B^{(t)}}{N} + \frac{\left(1 - B^{(t)}\right)}{N}\frac{\left(\mathbf{x}^{(t)} - \mathbf{c}^{(0)}\right)}{R}, \\
\rho\left(\mathbf{D}_t\right) &= \left(1 - \frac{1 - B^{(t)}}{N}\right)\rho\left(\mathbf{D}_{t-1}\right) + \frac{B^{(t)}}{N} + \frac{\left(1 - B^{(t)}\right)}{N}\frac{\left(\mathbf{x}^{(t)} - \mathbf{c}^{(0)}\right)^\top\left(\mathbf{x}^A - \mathbf{c}^{(0)}\right)}{R\left\|\mathbf{x}^A - \mathbf{c}^{(0)}\right\|}.
\end{aligned}$$

Next we use the linearity of $\mathrm{E}[\cdot]$ and the mutual independence of the random variables $B^{(t)}$ and $\mathbf{x}^{(t)}$ to compute the expectation of $\rho\left(\mathbf{D}_t\right)$. Importantly, $\mathbf{D}_{t-1}$ is also mutually independent of $B^{(t)}$ and $\mathbf{x}^{(t)}$ from the t^{th} iteration. Finally, using the fact that $\mathrm{E}\left[B^{(t)}\right] = \nu$ and $\mathrm{E}\left[\mathbf{x}^{(t)}\right] = \mathbf{c}^{(0)}$, we arrive at the following recursive formula $\mathrm{E}\left[\rho\left(\mathbf{D}_t\right)\right] = \left(1 - \frac{(1-\nu)}{N}\right)\mathrm{E}\left[\rho\left(\mathbf{D}_{t-1}\right)\right] + \frac{1}{N}\nu$. Unwrapping this recursion and using the

[3] In a more realistic model, benign data would not always be accepted, particularly once the attack had significantly shifted the detector. This would motivate the attacker to concentrate its attack mass in the early iterations. Amenable models for this scenario are further explored in Kloft & Laskov (2012).

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

facts that $\mathbf{D}_0 = \mathbf{0}$ and thus $\rho(\mathbf{D}_0) = 0$ yield the following geometric series:

$$\begin{aligned}
\mathrm{E}[\rho(\mathbf{D}_t)] &= \frac{\nu}{N} \cdot \sum_{t=1}^{T} \left(1 - \frac{(1-\nu)}{N}\right)^{T-t} \\
&= \frac{\nu}{N} \cdot \frac{1 - \left(1 - \frac{(1-\nu)}{N}\right)^{T}}{1 - \left(1 - \frac{(1-\nu)}{N}\right)} \\
&= \frac{\nu}{1-\nu} \left(1 - \left(1 - \frac{(1-\nu)}{N}\right)^{T}\right).
\end{aligned}$$

The upper bound of $\frac{\nu}{1-\nu}$ on this quantity is derived from the fact that, for all T, the last factor in the above expression is less than or equal to one. □

In addition to the above result, Kloft & Laskov (2012) also bound the variance of $\rho(\mathbf{D}_t)$ and show that it vanishes as $T, N \to \infty$. Thus, for sufficiently large N, the above formula for $\mathrm{E}[\rho(\mathbf{D}_t)]$ should accurately predict $\rho(\mathbf{D}_t)$ as an attack progresses. According to this result, Figure 4.6 depicts the number of iterations T relative to N that are predicted for mixed-data attacks as a function of the desired relative displacement D_R for various values of ν. As suggested by the bound in Theorem 4.22, for $\nu < 1$, displacements that exceed $D_R > \frac{\nu}{1-\nu}$ are not achievable regardless of the attack's duration T or the initial number of points, N. Further, since this bound strictly increases in ν we can invert it using Lemma 4.1, which suggests the adversary must control a fraction $\nu \geq \frac{D_R}{1+D_R}$ of the new data to expect to be able to achieve its goal.

These results are analogous to those of Section 4.4 where $1 - \nu$ plays a similar role to $\frac{N}{N+M}$ (see Figure 4.3). The absolute upper bounds are obtained under replacement by

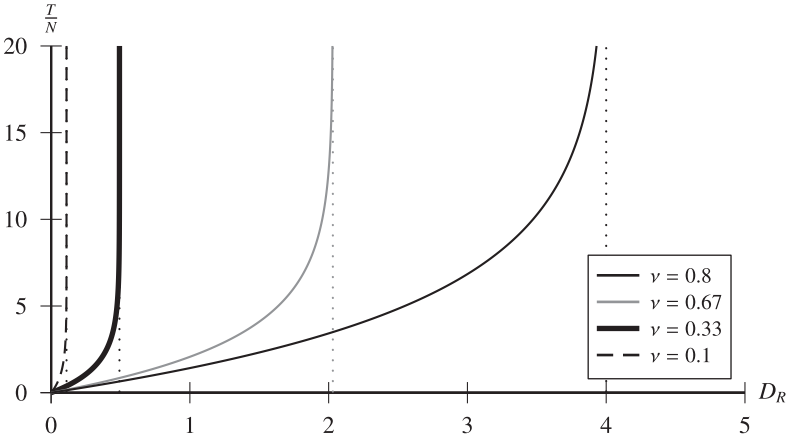


Figure 4.6 This plot show the theoretically predicted expected effect of the greedy optimal attacks of Theorem 4.22 for various values of the traffic mixture parameter: $\nu \in \{0.1, 0.33, 0.67, 0.8\}$. The plot shows the expected number of iterations T (relative to N) that are predicted for the mixed-data attacks as a function of the desired relative displacement D_R. The dotted lines depict the asymptotic maximum displacement that can be achieved for each ν.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

limiting the fraction of data controlled by the adversary, rather than limiting the attack duration. However, the results here just give the expected behavior rather than a strict bound on attack performance. In fact, in the worst case, all adversarial points could be accepted, resulting in the linear behavior of Section 4.5.1. Nonetheless, these results show that such results are generally overly pessimistic about the adversary's power. However, they also relied on assumptions about the benign data's distribution. Alternatively, Kloft & Laskov (2012) also examined a scenario in which the hypersphere would be manually reset if its false-positive rate becomes too high. In this alternative scenario, we no longer need to assume that benign data is always accepted for retraining, and under it, Kloft & Laskov (2012) derive a result similar to Theorem 4.22; however, for the sake of brevity, we will not explore that scenario here.

4.6.3 Extensions

There are several straightforward extensions of this work. The first extends the results to a hyper-ellipsoid detector defined by the Mahalanobis norm $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \mathbf{x}^\top \mathbf{\Sigma}^{-1} \mathbf{x}$ for a fixed positive-definite structure matrix $\mathbf{\Sigma}$. Under this norm, the hyper-ellipsoid detector is defined as $f_{\mathbf{c},\mathbf{\Sigma},R}(\mathbf{x}) =$ "+" if $\|\mathbf{x} - \mathbf{c}\|_{\mathbf{\Sigma}} > R$ and "−" otherwise. By transforming the problem into the space defined by $\mathbf{x}' \leftarrow \mathbf{\Sigma}^{-\frac{1}{2}} \mathbf{x}$ (which is possible since $\mathbf{\Sigma}$ is positive definite), all of the results of this chapter can be directly applied. The only caveat is that $\mathbf{\Sigma}$ distorts the space—hence the hardness of the task (given by D_R) depends on where the target point $\mathbf{x}^A$ is relative to the principal axes of $\mathbf{\Sigma}$.

A second extension involves hypersphere-based detection in an implicit feature space defined by a kernel function, which computes the inner product for data points implicitly projected into a Hilbert space $\mathcal{H}$. In particular, if $k : \mathcal{X} \times \mathcal{X} \to \Re$ is a kernel function and $\phi : \mathcal{X} \to \mathcal{H}$ is its corresponding projection function satisfying $k\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \phi\left(\mathbf{x}^{(1)}\right)^\top \phi\left(\mathbf{x}^{(1)}\right)$, then the centroid of the projected dataset is given by $\phi_C = \frac{1}{N} \sum_{i=1}^{N} \phi\left(\mathbf{x}^{(i)}\right)$ and the distance of the projected data point $\phi(\mathbf{x})$ from this centroid is

$$\|\phi(\mathbf{x}) - \phi_C\|_k = \left(k(\mathbf{x}, \mathbf{x}) - \frac{2}{N} \sum_{i=1}^{N} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) + \frac{1}{N^2} \sum_{i,j=1}^{N} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \right)^{\frac{1}{2}},$$

which, as with all kernel algorithms, can be computed implicitly only using the kernel function. The corresponding classification function labels the point $\mathbf{x}$ as "+" if $\|\phi(\mathbf{x})\|_k > R$ and as "−" otherwise.

Attacks against these kernel-based hypersphere detectors are a straightforward extension of the work presented above *if* we assume the attacker can insert arbitrary attack points directly in the feature space $\mathcal{H}$. However, a true adversary is restricted to inserting data points in the space $\mathcal{X}$, for which there is not generally a one-to-one mapping to $\mathcal{H}$. It is generally nontrivial for the adversary to find a point $\mathbf{a}^{(t)} \in \mathcal{X}$ whose image in feature space maximizes the displacement alignment according to Definition (4.2)—this is the well-known pre-image problem (Fogla & Lee 2006), which we also revisit in this book in other contexts (for example, see Section 8.4.3). Nonetheless, Kloft & Laskov

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

(2010) examined attacks against kernel-based hypersphere detectors empirically under the stronger assumption that the attacker could create its attack points in feature space—a conservative security assumption.

4.7 Summary

In this chapter, we analyzed *Causative Integrity* attacks against a hypersphere learner, which iteratively retrains its centroid based on new data. This analysis provides, under a variety of different assumptions, a deep understanding of the impact an attack can have on a simple learning model. We showed how optimal attacks can be constructed when assuming different powers for the adversary using several models for retraining and the outcomes demonstrate that the adversary's success critically depends on the scenario's assumptions. First, in Sections 4.3 and 4.4 we proved that, without any data replacement or time constraints, the attacker requires at least $M^\star \geq \exp(D_R - 1)$ attack points when $N = 0$ or $M^\star \geq N(\exp(D_R) - 1)$ when $N > 0$ to achieve the desired relative displacement of D_R. Similarly, if the attack has a maximum duration of T, these bounds increase to $M^\star \geq \left(\frac{T}{T - D_R}\right)^T$ when $N = 0$ or $M^\star \geq N\left(\frac{T}{T - D_R}\right)^T - N$ when $N > 0$ (assuming, in both cases, that $T > D_R$ because otherwise the desired displacement is unachievable). In all of these cases, the attacker requires *exponentially many attack points* in the size of its objective; i.e., the relative displacement, D_R.

However, bootstrap retraining without any data replacement severely limits the model's ability to adapt to data drift over time—eventually the model will become rigidly fixed even without an attack. Thus, in Sections 4.5 and 4.6, we revisit the data replacement settings analyzed by Kloft & Laskov (2012), in which each new data point replaces an old data point. These results show that under average-out and random-out replacement, the attacker only requires *linearly many attack points* (relative to the number of initial points, N) to achieve the desired goal, D_R. Even the nearest-out replacement policy, which was selected to limit the adversary's influence, empirically also exhibited linear-like behavior (except in low-dimensional spaces). These results showed that when the attacker controls all the new data in this scenario, it can successfully execute its attack with relatively little effort. Be that as it may, in many circumstances it is too conservative to assume that the attacker controls all new training data. Thus, in the final part of this chapter, we explored the work of Kloft & Laskov (2012) in examining a mixed-data scenario, in which the new data is drawn both from benign and malicious sources. Under the assumption that each new data point is malicious with probability ν and otherwise benign (and that all benign data is always used for retraining), the attacker must control a fraction $\nu \geq \frac{D_R}{1 + D_R}$ of the new data to expect to be able to achieve its goal. Further, the expected displacement of the attacker no longer increases linearly with the number of attack points; thus, under this more realistic setting, we see that the attacker cannot easily achieve its objective.

The analyses presented in this chapter demonstrate that the success of a poisoning attack against a iteratively retrained learner depends on several factors including how

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press

the learner restricts new training data and how much control the attacker has. However, the exact analysis provided in this chapter requires some assumptions that are not easily justified in practical settings and only apply to a relatively simple learning algorithm with bootstrap retraining. Nonetheless, these exact analyses provide interesting insights into the abstract problem of data poisoning and serve as a guide for less theoretical analysis of more complicated learning problems. This early work on contamination models heavily influenced our subsequent approach to the adversarial learning framework that we describe in the remainder of this book and was one of the first attempts to treat this problem as an adversarial game between a learner and an adversary.

https://doi.org/10.1017/9781107338548.004 Published online by Cambridge University Press