Corrigendum to “Learning rotations with little regret” September 7, 2010

Elad Hazan
IBM Almaden
650 Harry Road
San Jose, CA 95120
ehazan@cs.princeton.edu

Satyen Kale
Yahoo! Research
4301 Great America Parkway
Santa Clara, CA 95054
skale@yahoo-inc.com

Manfred K. Warmuth[*]
Department of Computer Science
UC Santa Cruz
manfred@cse.ucsc.edu

Abstract

There is an unfortunate error in our paper “Learning rotations with little regret” [HKW10] which appeared in COLT 2010. The sampling procedure for the noise matrix given in [HKW10] does not produce matrices with the right density. In this corrigendum, we describe the error, and give a correct sampling procedure. Unfortunately, even with the correct sampling procedure, the regret bound we get is $O(n\sqrt{T})$, which is weaker than the claimed regret bound of $O(\sqrt{nT})$ of the original paper [HKW10]. However, in this corrigendum we give a new algorithm based on Online Gradient Descent which obtains the optimal regret bound of $O(\sqrt{nT})$ for the online learning of rotations problem.

1 Description of Error

The error is in Section 3 of the paper, where we claim that the noise matrix $\mathbf{N}$ in Algorithm 1 of [HKW10] is drawn according to the following density:

$$d\mu(\mathbf{N}) \;\propto\; \exp(-\varepsilon\|\mathbf{N}\|_\star)\, d\mathbf{N}, \tag{C1}$$

where $d\mu(\mathbf{N})$ denotes the induced probability measure on $\mathbb{R}^{n\times n}$ by the sampling process and $d\mathbf{N} = dN_{11}dN_{12}\ldots dN_{nn}$ the Lebesgue (i.e. uniform) measure on $\mathbb{R}^{n\times n}$. This claim is incorrect. The sampling scheme described in Algorithm 1 does not choose $\mathbf{N}$ with probability density proportional to $\exp(-\varepsilon\|\mathbf{N}\|_\star) = \exp(-\varepsilon\sum_i \sigma_i)$ w.r.t. the Lebesgue measure $d\mathbf{N}$. This is because we sample $\mathbf{N}$ via its singular value decomposition, $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. To compute expectations over $\mathbf{N}$, we need to use the change of variables formula involving the determinant of the Jacobian matrix. We have the following change of variables formula (see [Ede05] Theorem 3):

$$d\mathbf{N} = \frac{1}{Z}\prod_{i<j}|\sigma_i^2-\sigma_j^2|\; d\sigma_1 d\sigma_2\ldots d\sigma_n(\mathbf{U}d\mathbf{U}^\top)^\wedge(\mathbf{V}d\mathbf{V}^\top)^\wedge,$$

where Z is the normalization constant, and the wedge-product $(\mathbf{U}d\mathbf{U}^\top)^\wedge$ denotes the volume form for the Haar measure on the orthogonal group $\mathcal{O}(n)$. The Haar measure is the unique (up to scaling) measure on the orthogonal group $\mathcal{O}(n)$ that is invariant under multiplication (from the left or the right) by an arbitrary orthogonal matrix. The current sampling scheme of [HKW10] samples $\mathbf{N}$ by selecting $\mathbf{U}$ and $\mathbf{V}$ from the Haar measure over $\mathcal{O}(n)$, and the diagonal elements of $\mathbf{\Sigma}$ i.i.d. from the exponential distribution $\varepsilon\exp(-\varepsilon\sigma)\, d\sigma$, and then setting $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. The density of the probability distribution (w.r.t. the product measure) of $(\mathbf{U},\mathbf{\Sigma},\mathbf{V})$ is then

$$\frac{1}{Z}\varepsilon^n\exp(-\varepsilon{\textstyle\sum_i}\sigma_i)\; d\sigma_1 d\sigma_2\ldots d\sigma_n(\mathbf{U}d\mathbf{U}^\top)^\wedge(\mathbf{V}d\mathbf{V}^\top)^\wedge,$$

which by the change of variables formula given above, leads to the following correct expression for the density (w.r.t. the Lebesgue measure) of the sampling distribution for $\mathbf{N}$ used by Algorithm 1:

$$d\mu(\mathbf{N}) \;=\; \frac{\varepsilon^n\exp(-\varepsilon\sum_i\sigma_i)}{Z\prod_{i<j}|\sigma_i^2-\sigma_j^2|}\, d\mathbf{N}.$$

[*]Supported by NSF grant IIS-0917397

2 Weaker Regret Bound via Correct Sampling Procedure

There is a way to efficiently sample the noise matrix $\mathbf{N}$ according density (C1). Since this is a log-concave density, the algorithms of Lovász-Vempala [LV06a] can be used to sample from it. There is a more efficient method as well: sample a matrix $\mathbf{M}$ from the unit ball $B_\star = \{\mathbf{N} : \|\mathbf{N}\|_\star \le 1\}$ [1] and a scale factor r from the $\texttt{Gamma}(n^2+1, 1/\varepsilon)$ distribution, and output $r\mathbf{M}$ (see Appendix A for details).

However, even if we correctly sample $\mathbf{N}$ according to density (C1), then we can only show a weaker regret bound of $O(n\sqrt{L})$, where L is the total loss of best rotation matrix in hindsight, rather than the claimed bound of $O(\sqrt{nL})$. The regret bound of [HKW10] relies on the key inequalities (2) and (3) of that paper. Inequality (2) of [HKW10] is now valid because with the correct sampling scheme, we do have $d\mu(\mathbf{N}) = \frac{1}{Z_\star}\exp(-\varepsilon\|\mathbf{N}\|_\star)\ d\mathbf{N}$, where $Z_\star$ is the normalization factor, and hence the inequalities in [HKW10] bounding $\frac{d\mu(\mathbf{N}-\mathbf{y}_t\mathbf{x}_t^\top)}{d\mu(\mathbf{N})} \ge 1-\varepsilon$ are correct.

However inequality (3) of [HKW10] is not valid and we can only replace it with the following weaker inequality which only gives the weaker regret bound of $O(n\sqrt{L})$:

$$\mathbf{E}[\mathbf{N} \bullet (\mathbf{R}_1 - \mathbf{R}^\star)] \le \frac{2n^2}{\varepsilon}, \tag{C2}$$

To see this, we first bound the LHS of (C2) by $2\mathbf{E}[\|\mathbf{N}\|_\star]$ (as done in [HKW10]) and then notice that in our new sampling scheme $\|\mathbf{N}\|_\star$ is distributed[2] as $\texttt{Gamma}(n^2, 1/\varepsilon)$ (see Appendix A), which implies that $\mathbf{E}[\|\mathbf{N}\|_\star] = n^2/\varepsilon$.

3 Optimal Regret via Online Gradient Descent

We now give a different algorithm for the rotation learning problem which obtains the optimal regret that was claimed in the original paper. The main idea is to run the Online Gradient Descent (OGD) algorithm (see e.g. Zinkevich [Zin03]). We first use this algorithm to learn an orthogonal matrix in an on-line fashion, i.e. we are allowed to predict with an orthogonal matrix rather than a rotation matrix. To be precise, in round t we are given a unit vector $\mathbf{x}_t$, and are required to produce an orthogonal matrix $\mathbf{U}_t$ and predict $\hat{\mathbf{y}}_t = \mathbf{U}_t\mathbf{x}_t$. Then a vector $\mathbf{y}_t$ is revealed, and we suffer the loss $L_t(\mathbf{U}_t) = \frac{1}{2}\|\mathbf{U}_t\mathbf{x}_t - \mathbf{y}_t\|^2 = 1 - (\mathbf{y}_t\mathbf{x}_t^\top) \bullet \mathbf{U}_t$..

The gradient of the loss $L_t(\mathbf{U}_t)$ with respect to $\mathbf{U}_t$ is $-\mathbf{y}_t\mathbf{x}_t^\top$. The OGD algorithm does a gradient descent step in each trial and then projects the resulting matrix into a convex set, here the convex hull of the set of all orthogonal matrices, $\mathcal{O}(n)$. Later, we show how to solve the rotations problem by modifying the OGD algorithm for learning orthogonal matrices.

We first characterize the convex hull of $\mathcal{O}(n)$:

Lemma 1 *The convex hull of $\mathcal{O}(n)$ is exactly the set of matrices of spectral norm at most 1: $B = \{\mathbf{M} : \|\mathbf{M}\| \le 1\}$. Furthermore, given any matrix $\mathbf{M} \in B$, there is a polynomial time randomized rounding algorithm, which produces a random orthogonal matrix $\widetilde{\mathbf{M}}$ such that $\mathbf{E}[\widetilde{\mathbf{M}}] = \mathbf{M}$.*

Proof: Since all singular values of any orthogonal matrix are equal to 1, we have $\mathcal{O}(n) \subseteq B$, and hence the convex hull of $\mathcal{O}(n)$ is contained in B. We now give the randomized rounding procedure, which automatically implies that B is contained in the convex hull of $\mathcal{O}(n)$, completing the characterization.

Let $\mathbf{M} \in B$ be any square matrix, and let $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the SVD of $\mathbf{M}$, where $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$. Since $\|\mathbf{M}\| \le 1$, the singular value $\sigma_i \in [0,1]$ for all $i = 1, 2, \ldots, n$. Consider the random diagonal matrix $\widetilde{\mathbf{\Sigma}} = \operatorname{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n)$ defined as

$$\tilde{\sigma}_i = \begin{cases} 1 & \text{with probability } (1+\sigma_i)/2 \\ -1 & \text{with probability } (1-\sigma_i)/2 \end{cases}$$

Note that for all i, we have $\mathbf{E}[\tilde{\sigma}_i] = \sigma_i$ and therefore $\mathbf{E}[\widetilde{\mathbf{\Sigma}}] = \mathbf{\Sigma}$. Furthermore, $\widetilde{\mathbf{\Sigma}}$ is an orthogonal matrix, and hence $\widetilde{\mathbf{M}} = \mathbf{U}\widetilde{\mathbf{\Sigma}}\mathbf{V}^\top$ is an orthogonal matrix too. Finally, by the linearity of the expectation, $\mathbf{E}[\widetilde{\mathbf{M}}] = \mathbf{M}$. ■

Gradient descent in our matrix setting is motivated by regularizing with half of the squared Frobenius norm, which is defined as $\|\mathbf{M}\|_F = \sqrt{\mathbf{M} \bullet \mathbf{M}} = \sqrt{\sum_{ij} M_{ij}^2}$. After a gradient descent step, the algorithm projects the parameter matrix back into the convex set B, where the projection is with respect to the same norm. It is easy to compute this projection via the SVD:

[1] This can also be done since the unit ball is a well-rounded convex body, with membership and separation oracles, see [LV06b] for details.

[2] For a slightly worse bound, we can simply bound $\|\mathbf{N}\|_\star$ by the scaling factor r chosen in the sampling procedure, since $\|\mathbf{M}\|_\star \le 1$, and using the fact that r is distributed as $\texttt{Gamma}(n^2+1, 1/\varepsilon)$.

Lemma 2 *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be any square matrix and let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the SVD of this matrix, where $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$. For all $i = 1, 2, \ldots, n$, let $\tilde{\sigma}_i = \min\{\sigma_i, 1\}$, and let $\widetilde{\mathbf{\Sigma}} = diag(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n)$. Then the projection of $\mathbf{M}$ onto B is*

$$\arg\min_{\mathbf{M}'\in B} \|\mathbf{M} - \mathbf{M}'\|_F = \mathbf{U}\widetilde{\mathbf{\Sigma}}\mathbf{V}^\top.$$

Proof: Since the spectral norm and Frobenius norm are unitarily invariant, it is easy to see that

$$\arg\min_{\mathbf{M}'\in B} \|\mathbf{M} - \mathbf{M}'\|_F \; = \; \mathbf{U}\left(\arg\min_{\mathbf{Q}\in B} \|\mathbf{\Sigma} - \mathbf{Q}\|_F\right)\mathbf{V}^\top.$$

So it suffices to focus on computing the projection for the diagonal matrix $\mathbf{\Sigma}$: We need to find a matrix $\mathbf{Q} \in B$ which minimizes $\sum_{ij}(\Sigma_{ij} - Q_{ij})^2$. Since $\|\mathbf{Q}\| \le 1$, we get that $Q_{ij} \in [-1, 1]$.

Consider a relaxation of the problem where we seek a matrix $\mathbf{Q}$ which minimizes $\sum_{ij}(\Sigma_{ij} - Q_{ij})^2$ subject only to the constraints $Q_{ij} \in [-1, 1]$. Since the off diagonal elements of $\mathbf{\Sigma}$ are zero, the solution of the relaxed problem is a diagonal matrix. Furthermore, since all $\Sigma_{ii} = \sigma_i \in [0, 1]$, the i-th diagonal element of the solution is $\tilde{\sigma}_i = \min\{\sigma_i, 1\}$. Hence, the optimal solution to the relaxed problem is $\widetilde{\mathbf{\Sigma}} = \text{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n)$. This matrix lies in B as well since all $\tilde{\sigma}_i \in [0, 1]$. Hence $\widetilde{\mathbf{\Sigma}}$ is also the optimal solution for the original projection problem over B. ■

Now, we can describe the OGD algorithm over B. This requires a parameter η which we tune later.

Algorithm 1 OGD on B

```
1: Let $\mathbf{W}_1$ be an arbitrary orthogonal matrix (for instance, the identity matrix $I$).
2: for $t = 1$ to $T$ do
3:     Obtain vector $\mathbf{x}_t$.
4:     Randomly round $\mathbf{W}_t$ to an orthogonal matrix $\mathbf{U}_t$ and use it in round $t$.
5:     Predict $\hat{\mathbf{y}}_t = \mathbf{U}_t\mathbf{x}_t$ and observe the result vector $\mathbf{y}_t$. Suffer loss $\frac{1}{2}\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|^2$.
6:     Update $\mathbf{W}'_{t+1} = \mathbf{W}_t + \eta\, \mathbf{y}_t\mathbf{x}_t^\top$.
7:     Compute the projection $\mathbf{W}_{t+1}$ of $\mathbf{W}'_{t+1}$ onto $B$.
8: end for
```

Note that in expectation, we predict using the matrix $\mathbf{W}_t$ at round t, and hence the expected regret can be bounded by the regret for playing $\mathbf{W}_t$. The following elementary regret bound uses squared Frobenius norm as a measure of progress (see e.g. [CBLW96]) and the Pythagorean Theorem to show that the projections step does not hurt (see [HW01]).

Theorem 3 *If Algorithm 1 is run with $\eta = 2\sqrt{\frac{n}{T}}$, then it produces random orthogonal matrices $\mathbf{U}_t$ such that*

$$\mathbf{E}[\mathit{Regret}_T] \; = \; \sum_{t=1}^{T} \mathbf{E}[L_t(\mathbf{U}_t)] - \min_{\mathbf{U}\in\mathcal{O}(n)} \sum_{t=1}^{T} L_t(\mathbf{U}) \; \le \; 2\sqrt{nT}.$$

Proof: For any comparator $\mathbf{U} \in B$,

$$\begin{aligned} &\|\mathbf{W}_{t+1} - \mathbf{U}\|_F^2 = \|\text{proj}(\mathbf{W}_t + \eta\mathbf{y}_t\mathbf{x}_t^\top) - \mathbf{U}\|_F^2 \\ &\le \quad \|\mathbf{W}_t + \eta\, \mathbf{y}_t\mathbf{x}_t^\top - \mathbf{U}\|_F^2 = \|\mathbf{W}_t - \mathbf{U}\|_F^2 + 2\eta(\mathbf{W}_t - \mathbf{U}) \bullet (\mathbf{y}_t\mathbf{x}_t^\top) + \eta^2 \underbrace{\|\mathbf{y}_t\mathbf{x}_t^\top\|_F^2}_{=1}, \end{aligned}$$

where the inequality follows from the Pythagorean Theorem (see [HW01] for an extended discussion of this). By rearranging we get the following inequality for each trial:

$$(1 - \mathbf{W}_t \bullet (\mathbf{y}_t\mathbf{x}_t^\top)) - (1 - \mathbf{U} \bullet (\mathbf{y}_t\mathbf{x}_t^\top)) \; \le \; \frac{\|\mathbf{W}_t - \mathbf{U}\|_F^2 - \|\mathbf{W}_{t+1} - \mathbf{U}\|_F^2}{2\eta} + \frac{\eta}{2}.$$

By summing over all T trials we get

$$\sum_t L_t(\mathbf{W}_t) - \sum_t L_t(\mathbf{U}) \; \le \; \frac{\|\mathbf{W}_1 - \mathbf{U}\|_F^2 - \|\mathbf{W}_{T+1} - \mathbf{U}\|_F^2}{2\eta} + \frac{\eta T}{2} \; \le \; \frac{2n}{\eta} + \frac{\eta T}{2},$$

since $\|\mathbf{W}_1 - \mathbf{U}\|_F^2 \le n\|\mathbf{W}_1 - \mathbf{U}\|^2 \le 2n\|\mathbf{W}_1\|^2 + 2n\|\mathbf{U}\|^2 \le 4n$ and $\|\mathbf{W}_{T+1} - \mathbf{U}\|_F^2 \ge 0$. The RHS is minimized at $2\sqrt{nT}$ by choosing $\eta = 2\sqrt{\frac{n}{T}}$. ■

The running time is dominated by computing the SVD of $\mathbf{W}_t$, which is $O(n^3)$.

3.1 Solving the Online Learning of Rotations Problem

We now solve the online learning of rotations problem via a special trick, that exploits the fact that we are given the vector $\mathbf{x}_t$ before we are required to produce the rotation matrix $\mathbf{R}_t$. We run Algorithm 1, except for the following addition to line 4: if the random orthogonal matrix $\mathbf{U}_t$ happens to be a rotation matrix, then we set $\mathbf{R}_t = \mathbf{U}_t$. Otherwise, we set $\mathbf{R}_t$ to be an arbitrary rotation matrix such that $\mathbf{R}_t\mathbf{x}_t = \mathbf{U}_t\mathbf{x}_t$. Note that this can be done since we can always rotate any unit vector into any other unit vector (an explicit construction is given at the end of this subsection). Call the resulting algorithm for the online learning of rotations problem Algorithm 2.

Algorithm 2 predicts with $\hat{\mathbf{y}}_t = \mathbf{R}_t\mathbf{x}_t$, which is equal $\mathbf{U}_t\mathbf{x}_t$. Hence $\mathbf{E}[L_t(\mathbf{R}_t)] = \mathbf{E}[L_t(\mathbf{U}_t)]$ and from Theorem 3, we immediately get the following regret bound:

Theorem 4 *If Algorithm 2 is run with $\eta = 2\sqrt{\frac{n}{T}}$, then it produces random rotation matrices $\mathbf{R}_t$ such that*

$$\sum_{t=1}^{T} \mathbf{E}[L_t(\mathbf{R}_t)] - \min_{\mathbf{U}\in\mathcal{O}(n)} \sum_{t=1}^{T} L_t(\mathbf{U}) \;\leq\; 2\sqrt{nT}.$$

Algorithm also runs in $O(n^3)$ time per iteration. Note that this is actually a stronger regret bound, since we measure regret against the best *orthogonal matrix*, rather than the best rotation matrix and $\mathcal{O}(n) \supseteq \mathcal{SO}(n)$.

Finally, we give an explicit construction of a rotation matrix that takes a given unit vector $\mathbf{x}$ into any other given unit vector $\mathbf{y}$. Let $\mathbf{U}$ be an orthogonal matrix whose first column is $\mathbf{x}$ and whose second column $\mathbf{z}$ is chosen so that the span of $\{\mathbf{x}, \mathbf{z}\}$ equals the span of $\{\mathbf{x}, \mathbf{y}\}$. Clearly $\mathbf{U}$ can be found by starting the Gram-Schmidt orthonormalization with the vectors $\{\mathbf{x}, \mathbf{y}\}$ and completing it to a basis of the whole space. Let $\mathbf{G}$ be the following Givens rotation matrix: $G_{11} = G_{22} = \mathbf{y}\cdot\mathbf{x}$, $G_{21} = -G_{12} = \mathbf{y}\cdot\mathbf{z}$, $G_{ii} = 1$ for all $i > 2$, and all other entries are 0. Then the matrix $\mathbf{R} = \mathbf{U}\mathbf{G}\mathbf{U}^\top$ is a rotation matrix since $\mathbf{U}$ is an orthogonal and $\mathbf{G}$ a rotation matrix. Furthermore, it is easy to check that $\mathbf{R}\mathbf{x} = (\mathbf{y}\cdot\mathbf{x})\,\mathbf{x} + (\mathbf{y}\cdot\mathbf{z})\,\mathbf{z} = \mathbf{y}$.

4 A final question

Even though the regret bound of $O(\sqrt{nT})$ matches the lower bound given in [HKW10], it is still slightly weaker than the claimed bound of [HKW10], which was $O(\sqrt{nL})$, where L is the loss of the best rotation matrix for the given example sequence. The vanilla OGD analysis doesn't give the tighter bound. Obtaining the tighter bound remains an open problem.

References

[CBLW96] N. Cesa-Bianchi, P. M. Long, and M. K. Warmuth. Worst-case quadratic loss bounds for on-line prediction of linear functions by gradient descent. *IEEE Transactions on Neural Networks*, 7(3):604–619, 1996. Earlier version in 6th COLT, 1993.

[Ede05] Alan Edelman. Jacobians of matrix transforms (with wedge products). *(Class notes) 18.325: Finite Random Matrix Theory*, 2005. http://web.mit.edu/18.325/www/handouts/handout3.pdf.

[HKW10] Elad Hazan, Satyen Kale, and Manfred K. Warmuth. Learning rotations with little regret. In *COLT*, 2010.

[HW01] M. Herbster and M. K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

[LV06a] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *FOCS*, pages 57–68, 2006.

[LV06b] László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM J. Comput.*, 35(4):985–1005, 2006.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

A Sampling

We describe a general procedure to sample points in $\mathbb{R}^d$ from a distribution $\mathcal{D}$, with density (w.r.t. the Lebesgue measure) proportional to $\exp(-\varepsilon\|\mathbf{x}\|)\ d\mathbf{x}$ for some norm $\|\cdot\|$ on $\mathbb{R}^d$. In the paper, we apply this sampling procedure with the domain being matrices in $\mathbb{R}^{n\times n}$, and the norm being the trace norm $\|\cdot\|_\star$ for matrices, being careful to note that the dimension $d = n^2$.

We need some notation first. Let B be the unit ball in $\|\cdot\|$, i.e. $B = \{\mathbf{x} \in \mathbb{R}^d : \ \|\mathbf{x}\| \leq 1\}$, and let ∂B be its boundary, i.e. $\partial B = \{\mathbf{x} \in \mathbb{R}^d : \ \|\mathbf{x}\| = 1\}$. We now give an efficient algorithm to sample from $\mathcal{D}$, assuming we can sample a point uniformly at random from B.

Sampling from $\mathcal{D}$.

1. Choose r from the $\texttt{Gamma}(d+1, 1/\varepsilon)$ distribution.
2. Choose $\mathbf{v} \in B$ uniformly at random.
3. Output $r\mathbf{v}$.

Theorem 5 *The given procedure samples* $\mathbf{x}$ *with density proportional to* $\exp(-\varepsilon\|\mathbf{x}\|)\ d\mathbf{x}$.

Proof: The sampling procedure given above can be equivalently described as choosing $r \sim \texttt{Gamma}(d + 1, 1/\varepsilon)$ and then sampling a point uniformly at random from rB. A given point $\mathbf{x}$ will be sampled only if $r \geq \|\mathbf{x}\|$. Define $V(r) = \text{Vol}(rB) = r^d V(1)$. Then the density at point $\mathbf{x}$ can be obtained as follows:

$$\begin{aligned}\int_{r=\|\mathbf{x}\|}^{\infty} \frac{d\mathbf{x}}{V(r)} \cdot \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr &= \int_{r=\|\mathbf{x}\|}^{\infty} \frac{1}{r^d V(1)} \cdot \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr d\mathbf{x} \\ &= -\frac{1}{\varepsilon^{-d} V(1)\Gamma(d+1)} e^{-\varepsilon r} d\mathbf{x}\Big|_{\|\mathbf{x}\|}^{\infty} \\ &= \frac{e^{-\varepsilon\|\mathbf{x}\|}}{\varepsilon^{-d} V(1)\Gamma(d+1)} d\mathbf{x}.\end{aligned}$$

■

The following gives a characterization of the distribution of $\|\mathbf{x}\|$:

Lemma 6 *The distribution of* $\|\mathbf{x}\|$ *when* $\mathbf{x}$ *is sampled from* $\mathcal{D}$ *is* $\texttt{Gamma}(d, 1/\varepsilon)$.

Proof: The given sampling procedure chooses a real number $r \geq 0$ from the $\texttt{Gamma}(d+1, 1/\varepsilon)$ distribution and a vector $\mathbf{v} \in B$ uniformly at random. Thus, by the Law of Total Probability, we have

$$\begin{aligned}\mathbf{Pr}[\|\mathbf{x}\| \leq R] &= \int_{r=0}^{\infty} \mathbf{Pr}[\mathbf{v} \in (R/r)B] \cdot \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr \\ &= \int_{r=0}^{R} 1 \cdot \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr + \int_{r=R}^{\infty} (R/r)^d \cdot \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr \\ &= \int_{r=0}^{R} \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr + \frac{R^d e^{-\varepsilon R}}{\varepsilon^{-d}\Gamma(d+1)}.\end{aligned}$$

Since

$$\frac{d}{dr}[r^d e^{-\varepsilon r}] = dr^{d-1}e^{-\varepsilon r} - \varepsilon r^d e^{-r},$$

by using integration by parts, we have

$$\begin{aligned}\mathbf{Pr}_{r\sim\texttt{Gamma}(d,1/\varepsilon)}[r \leq R] &= \int_{r=0}^{R} \frac{r^{d-1}e^{-\varepsilon r}}{\varepsilon^{-d}\Gamma(d)} dr \\ &= \frac{\varepsilon}{d}\int_{r=0}^{R} \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-d}\Gamma(d)} dr + \frac{1}{d}\int_{r=0}^{R} d\left[\frac{r^d e^{-\varepsilon r}}{\varepsilon^{-d}\Gamma(d)}\right] \\ &= \int_{r=0}^{R} \frac{r^d e^{-\varepsilon r}}{\varepsilon^{-(d+1)}\Gamma(d+1)} dr + \frac{R^d e^{-\varepsilon R}}{\varepsilon^{-d}\Gamma(d+1)}.\end{aligned}$$

This is the cumulative distribution function of the $\texttt{Gamma}(d, 1/\varepsilon)$ distribution, as required. ■