Efficiently sampling vectors and coordinates from the n-sphere and n-ball

Aaron R. Voelker, Jan Gosmann, Terrence C. Stewart
Centre for Theoretical Neuroscience – Technical Report

January 4, 2017

Abstract

We provide a short proof that the uniform distribution of points for the n-ball is equivalent to the uniform distribution of points for the $(n+1)$-sphere projected onto n dimensions. This implies the surprising result that one may uniformly sample the n-ball by instead uniformly sampling the $(n+1)$-sphere and then arbitrarily discarding two coordinates. Consequently, any procedure for sampling coordinates from the uniform $(n+1)$-sphere may be used to sample coordinates from the uniform n-ball without any modification. For purposes of the Semantic Pointer Architecture (SPA), these insights yield an efficient and novel procedure for sampling the dot-product of vectors—sampled from the uniform ball—with unit-length encoding vectors.

1 Introduction

The Semantic Pointer Architecture (SPA; Eliasmith, 2013) is a cognitive architecture that has been used to model what still remains the world's largest functioning model of the human brain (Eliasmith et al., 2012). Core to the SPA is the notion of a *semantic pointer*, which is a high-dimensional vector that represents compressed semantic information. Consequently, the current compiler for the SPA (Nengo; Bekolay et al., 2013) makes extensive use of computational procedures for uniformly sampling vectors, either from the surface of the unit *n-sphere* ($\left\{\mathbf{s} \in \mathbb{R}^{n+1} : \|\mathbf{s}\| = 1\right\}$) or from the interior of the unit *n-ball* ($\{\mathbf{b} \in \mathbb{R}^{n} : \|\mathbf{b}\| < 1\}$). Furthermore, when building specific models, we sometimes sample the dot-product of these vectors with arbitrary unit-length vectors (Knight et al., 2016). In summary, the SPA requires efficient algorithms for uniformly sampling high-dimensional vectors and their coordinates (Gosmann and Eliasmith, 2016).

To begin, it is worth stating a few facts. We use the term 'coordinate' to refer to an element of some vector with respect to some basis. For uniformly distributed vectors from the n-ball or n-sphere, the choice of basis for the coordinate system is arbitrary (and need not even stay fixed between samples) – but it is helpful to consider the standard basis. Relatedly, the dot-product of two vectors sampled uniformly from the n-sphere is equivalent to the distribution of any coordinate of a vector sampled uniformly from the n-sphere. Similarly, the dot-product of a vector sampled uniformly from the n-ball with a vector sampled uniformly from the n-sphere is equivalent to the distribution of any coordinate of a vector sampled uniformly from the n-ball. These last two facts hold simply because we may suppose one of the unit vectors is elementary after an appropriate change of basis, in which case their dot-product extracts the corresponding coordinate.

Now there exist well-known algorithms for sampling points (i.e., vectors) from the n-sphere and n-ball. We review these in sections §2.1 and §2.2 respectively. In §2.3 we briefly review how to efficiently sample coordinates from the uniform n-sphere. Our main contribution is a proof in §3 that the n-ball may be uniformly sampled by arbitrarily discarding two coordinates from the $(n+1)$-sphere. This result was previously discovered by Harman and Lacko (2010), specifically by setting $k = 2$ in Corollary 1 and working through some details. We derived this result independently and thus present it here in an explicit and self-contained manner. This leads to the development of two algorithms: in §3.1 we provide an alternative algorithm for uniformly sampling points from the n-ball, and in §3.2 we provide an efficient and novel algorithm for sampling coordinates from the uniform n-ball by a simple reduction to the $(n+1)$-sphere.

2 Preliminaries

To help make this a self-contained reference, we summarize some previously known results:

2.1 Uniformly sampling the n-sphere

To uniformly sample points from the unit n-sphere, defined as $\left\{\mathbf{s} \in \mathbb{R}^{n+1} : \|\mathbf{s}\| = 1\right\}$:

1. Independently sample $n+1$ normally distributed variables: $x_1, \ldots, x_{n+1} \sim \mathcal{N}(0,1)$.[1]
2. Compute their ℓ_2-norm: $r = \sqrt{\sum_{i=1}^{n+1} x_i^2}$.
3. Return the vector $\mathbf{s} = (x_1, \ldots, x_{n+1})/r$.

This is implemented in Nengo as `nengo.dists.UniformHypersphere(surface=True)` with dimensionality parameter $d = n + 1$.

2.2 Uniformly sampling the n-ball

To uniformly sample points from the unit n-ball—defined as $\{\mathbf{b} \in \mathbb{R}^n : \|\mathbf{b}\| < 1\}$—we use the previous algorithm as follows:

1. Sample $\mathbf{s} \in \mathbb{R}^n$ from the $(n-1)$-sphere.
2. Uniformly sample $c \sim U[0,1]$.
3. Return the vector $\mathbf{b} = c^{1/n}\mathbf{s}$.

This is implemented in Nengo as `nengo.dists.UniformHypersphere(surface=False)` with dimensionality parameter $d = n$.

2.3 Uniformly sampling coordinates from the n-sphere

To sample coordinates from the unit n-sphere (i.e., uniform points from the sphere projected onto an arbitrary unit vector) we could simply modify §2.1 to return only a single element – but this would be inefficient for large n. Instead, we use `nengo.dists.CosineSimilarity`$(n+1)$ to directly sample the underlying distribution, via its probability density function (Voelker and Eliasmith, 2014; eq. 11):

$$f(x) \propto \left(1 - x^2\right)^{\frac{n}{2}-1},$$

which may be expressed using the "SqrtBeta" distribution (Gosmann and Eliasmith, 2016).[2]

3 Results

Lemma 1. *Let n be a positive integer, $x_1, \ldots, x_{n+2} \sim \mathcal{N}(0,1)$ be independent and normally distributed random variables, then:*[3]

$$c^{1/n} \overset{\text{D}}{=} \frac{\sqrt{\sum_{i=1}^{n} x_i^2}}{\sqrt{\sum_{i=1}^{n+2} x_i^2}}, \tag{1}$$

where $c \sim U[0,1]$ is a uniformly distributed random variable.

Proof. Let $X = \sum_{i=1}^{n} x_i^2$ and $Y = \sum_{i=n+1}^{n+2} x_i^2$. Observe that $X \sim \chi^2(n)$, $Y \sim \chi^2(2)$, and $X \perp\!\!\!\perp Y$ (i.e., X and Y are independent chi-squared variables with n and 2 degrees of freedom, respectively). Using relationships between the chi-squared/Beta/Kumaraswamy distributions, we know that:

$$\frac{X}{X+Y} \sim \beta(n/2, 1) \implies \frac{X}{X+Y} \sim \texttt{Kumaraswamy}\,(n/2, 1) \implies \left(\frac{X}{X+Y}\right)^{n/2} \sim U[0,1].$$

Focusing on the final distribution, raise both sides to the exponent $1/n$ to obtain (1). ☐

[1] The choice of variance for the normal distribution is an arbitrary constant.

[2] https://github.com/nengo/nengo/blob/614e7657afd1f16b296a06068f3d4673e5b575d2/nengo/dists.py#L431

[3] We use $\overset{\text{D}}{=}$ to denote that two random variables have the same distribution.

Theorem 1. *Let n be a positive integer, $\mathbf{b}$ be a random n-dimensional vector uniformly distributed on the unit n-ball, $\mathbf{s}$ be a random $(n+2)$-dimensional vector uniformly distributed on the unit $(n+1)$-sphere, and finally $\mathbf{P} \in \mathbb{R}^{n,n+2}$ be any rectangular orthogonal matrix,*[4] *then:*

$$\mathbf{b} \overset{\text{D}}{=} \mathbf{P}\mathbf{s}. \tag{2}$$

Proof. By §2.1, $\mathbf{s} = (x_1, \ldots, x_{n+2})/r$, where $x_1, \ldots, x_{n+2} \sim \mathcal{N}(0,1)$ and $r = \sqrt{\sum_{i=1}^{n+2} x_i^2}$. Also let $\tilde{r} = \sqrt{\sum_{i=1}^{n} x_i^2}$. Since the uniform distribution for the sphere (and for the ball) is isomorphic under change of basis, we may assume without loss of generality that $\mathbf{P}$ is the $(n+2)$-dimensional identity with its last two rows removed:

$$\begin{aligned}
\mathbf{P}\mathbf{s} &\overset{\text{D}}{=} (x_1, \ldots, x_n)/r \\
&= (\tilde{r}/r)(x_1, \ldots, x_n)/\tilde{r} \\
&\overset{\text{D}}{=} c^{1/n}(x_1, \ldots, x_n)/\tilde{r} && \text{(where } c \sim U[0,1] \text{ by Lemma 1)} \\
&\overset{\text{D}}{=} \mathbf{b} && \text{(by §2.2).}
\end{aligned}$$

□

3.1 Uniformly sampling the n-ball (alternative)

As a corollary to Theorem 1, we obtain the following alternative to §2.2 for the n-ball:

1. Sample $\mathbf{s} \in \mathbb{R}^{n+2}$ from the $(n+1)$-sphere.
2. Return the vector $\mathbf{b} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$.

3.2 Uniformly sampling coordinates from the n-ball

To efficiently sample coordinates from the uniform n-ball (i.e., uniform points from the ball projected onto an arbitrary unit vector), observe that in §3.1 the elements of $\mathbf{b}$ correspond directly to elements of $\mathbf{s}$. In other words, sampling coordinates from the uniform n-ball reduces to sampling coordinates from the uniform $(n+1)$-sphere. Therefore, we simply reuse the method from §2.3 to sample coordinates from the $(n+1)$-sphere: `nengo.dists.CosineSimilarity`$(n+2)$.

References

Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron R Voelker, and Chris Eliasmith. Nengo: a Python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7, 2013.

Chris Eliasmith. *How to build a brain: A neural architecture for biological cognition*. Oxford University Press, 2013.

Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *science*, 338(6111):1202–1205, 2012.

Jan Gosmann and Chris Eliasmith. Optimizing semantic pointer representations for symbol-like processing in spiking neural networks. *PLOS ONE*, 11(2):e0149928, 2016.

Radoslav Harman and Vladimír Lacko. On decompositional algorithms for uniform sampling from n-spheres and n-balls. *Journal of Multivariate Analysis*, 101(10):2297–2304, 2010.

James Knight, Aaron R Voelker, Andrew Mundy, Chris Eliasmith, and Steve Furber. Efficient SpiNNaker simulation of a heteroassociative memory using the Neural Engineering Framework. In *The 2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016.

Aaron R Voelker and Chris Eliasmith. Controlling the Semantic Pointer Architecture with deterministic automata and adaptive symbolic associations. Technical report, Centre for Theoretical Neuroscience, Waterloo, ON, 2014.

[4] We use "rectangular orthogonal" to mean $\mathbf{P}\mathbf{P}^\intercal = \mathbf{I}$ in this case, or equivalently the rows of $\mathbf{P}$ are orthonormal. This transformation matrix can be understood as a change of basis followed by the deletion of two coordinates.