Technical Note: Computing moments of a truncated Gaussian for EP in high-dimensions

Marc Toussaint
Machine Learning & Robotics group, TU Berlin
Franklinstr. 28/29, FR 6-9, 10587 Berlin, Germany

October 16, 2009

In this technical note we derive an algorithm for computing the moments of a truncated Gaussian in high-dimensions. In principle, all of this is well known and not novel. Herbrich has already an (unpublished) technical note on EP with truncated Gaussians at research.microsoft.com/pubs/74554/EP.pdf. However, getting an efficient algorithm in high-dimension is not so trivial. We derive one in this note. The corresponding source code is available at user.cs.tu-berlin.de/~mtoussai/source-code/.

Our motivation is the application in the context of Approximate Inference Control (6), where we use approximate inference to compute trajectories under hard constraints: Collision and joint avoidance implement messages of the form of heavyside functions; using Expectation Propagation with truncated Gaussians we can approximate the motion posterior.

1 1D case

Let us first address the simple 1D case. The problem is defined as follows: Let $x \in \mathbb{R}$ and $g(x) = e^{-x^2/2}$ and $\theta(x) = [x \geq z]$ (the heavyside function at $z$). We want to compute a Gaussian approximation of $g(x)\theta(x)$. For this we need to compute the moments of $g(x)\theta(x)$. For the norm (0th) moment we have:

$$
\int_0^\infty e^{-x^2} \, dx = \frac{\sqrt{\pi}}{2} \text{erf}(z)
$$

$$
M_0 := \int_0^\infty e^{-x^2} \, dx = \sqrt{2} \int_{z/\sqrt{2}}^\infty e^{-u^2} \, du = \sqrt{\pi/2} \left[1 - \text{erf}(z/\sqrt{2})\right]
$$

For the 1st moment:

$$
M_1 := \int_{z/\sqrt{2}}^{\infty} e^{-x^2/2} \, dx = -\left[ e^{t^2/2} \right]_{-\infty}^{-z^2/2} = -\left[ 0 - e^{-z^2/2} \right] = e^{-z^2/2}
$$

For the $n$-th moment:

$$
I_n(z, a) := \int_z^\infty e^{-ax^2} \, dx
$$

$$
\frac{\partial}{\partial a} I_n(z, a) = \int_z^\infty e^{-ax^2} (-x^2) \, dx = -I_{n+2}(z, a)
$$

(5)

$$
\frac{\partial}{\partial a} \text{erf}(\sqrt{a}z) = \frac{\text{erf}(\sqrt{a}z)}{2} - \frac{\sqrt{\pi} e^{-a z^2}}{\sqrt{a}}
$$

(6)

And hence for 2nd moment:

$$
I_2(z, a) = -\frac{\partial}{\partial a} N(z, a)
$$

(7)

$$
= a^{-3/2} \frac{\sqrt{\pi}}{4} \left[1 - \text{erf}(\sqrt{a}z)\right] + a^{-1/2} \frac{\sqrt{\pi}}{\sqrt{a}} \left[\frac{a^{-1/2}z}{\sqrt{\pi}} e^{-a z^2}\right]
$$

(8)

$$
M_2 := \int_z^\infty e^{-x^2/2} \, dx = I_2(z, 1/2)
$$

$$
= \sqrt{\pi/2} \left[1 - \text{erf}(z/\sqrt{2})\right] + z e^{-z^2/2}
$$

(9)

$$
M_2 = M_0 + z M_1
$$

(10)

In summary, we have

- norm $n := M_0 = \sqrt{\pi/2} \left[1 - \text{erf}(z/\sqrt{2})\right]$
- mean $m := M_1/M_0 = e^{-z^2/2}/n$
- variance $v := M_2/M_0 - m^2 = 1 + zm - m^2$

2 General case

We now have a $n$-dim Gaussian $f(y)$ and heavyside function $\theta(y)$ along a hyperplane with normal $c$ and offset $d$,

$$
f(y) \propto \exp\{-\frac{1}{2}(y-a)^\top A^1 (y-a)\}
$$

(15)

$$
\theta(y) = [\|c^\top y - d\| \geq 0]
$$

(16)

where $[\cdot]$ is the indicator function. We transform this problem such that the Gaussian becomes a standard Gaussian and the constraint is aligned with the $x$-axis.

We need two transformations for this: first a linear transformation to standardize the Gaussian, then a rotation to align with the $x$-axis. Let $A = M^\top M$ be the Cholesky decomposition ($A^1 = M^1 M^{-1}$) and we define $x = M^{-\top}(y-a)$. We have

$$
f(x) = \exp\{-\frac{1}{2} x^\top x\}
$$

(17)
Algorithm 1 Truncated Standard Gaussian

1: **Input:** $z$
2: **Output:** norm $n$, mean $m$, variance $v$
3: $n = \frac{\sqrt{\pi/2}(1 - \text{erf}(z/\sqrt{2}))}{2}$
4: $m = \exp(-z^2/2)/n$
5: $v = 1 + zm - m^2$

Algorithm 2 Truncate Gaussian

1: **Input:** mean $a$, covariance $A$, constraint coeffs $c, d$
2: **Output:** mean $b$, covariance $B$
3: $M^T M = A$ // Cholesky decomposition
4: $z = (c^T a + d)/|Mc|$ // as in equation (??)
5: $v = Mc/|Mc|$ // in equation (??)
6: $R = \text{rotation onto } v$ // as in equation (??)
7: $(m, v) = \text{Truncated Standard Gaussian}(z)$
8: $b = M^T Rb' + a$
9: $B = M^T Rdiag(v, 1, ..., 1) R^T$ M

Note that we defined $v$ to be normalized. (If $|Mc|$ is zero the truncation has no effect or zero likelihood, depending on whether $c^T a - d > 0$ or $c^T a - d < 0$, respectively.) We compute a rotation matrix that rotates the unit vector $e = (1, 0, \ldots, 0)$ onto $v$ (implemented in array.cpp). We define $x' = R^{-1} x$. We have $v = Re$ and

$$f(x') = \exp\left\{-\frac{1}{2} x'^T x'\right\}$$

$$\theta(x') = [(c^T (M^T x + a - d) \geq 0)] = [(v^T x + z \geq 0)]$$

That is, $\theta(x')$ truncates along the first axis in the $x'$ coordinate system. Given the mean $m$ and variance $v$ of the $(-z)$-truncated standard Gaussian, we have

$$f(x') \theta(x') \approx N(x'|b', B')$$

$$b' = (m, 0, \ldots, 0)$$

$$B' = \text{diag}(v, 1, \ldots, 1)$$

We undo the transformation $x' = R^T M^{-T}(y - a)$ and get the result

$$f(y) \theta(y) \approx N(y|b, B)$$

$$b = M^T Rb' + a$$

$$B = M^T R B' R^T M$$

which gives the mean and covariance of the truncated Gaussian. The explicit algorithms are given below. The figure illustrates the result of truncating a Gaussian in 2D with the constraint $[|x > 1|]$. 

![Figure 1: Truncation of a Gaussian at the constraint $[|x > 1|]$ in 2D.](image-url)