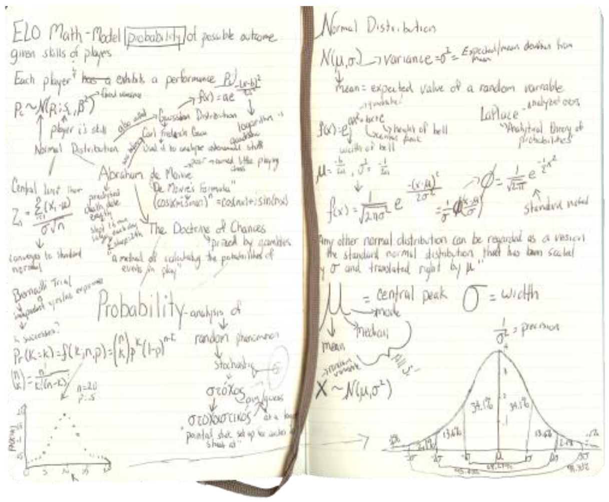
The Math Behind TrueSkill

Abstract

This paper accompanies my "Computing Your Skill" blog post at moserware.com. It contains selected portions from my paper notebook that I kept on my several-month journey to understand the TrueSkill algorithm. This paper is woefully incomplete, but hopefully is better than nothing.



Most math papers error on the side of being too terse with derivations; this approach makes it sometimes hard to follow how the author got from step to step. In this paper, I took the opposite approach and erred on the side of being explicit at the expense of using extra space.

I created a general order of concepts in this paper, but you're welcome to skip around as you see fit.

Prerequisites

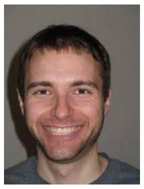
This paper assumes that you have had some exposure to math through the calculus level. In addition, it helps if you have had exposure with matrices and statistics.

I tried to help with some prerequisites by adding hyperlinks to refreshers on the basics (mainly to Wikipedia entries).

Version

This paper was last edited on **5/28/2011**. I will plan to update it as needed based on questions or further research. Feel free to send in suggestions for updates.

Author & License



Contents

Notation

Symbol	Meaning
$\Phi(\mathrm{x})$	Cumulative distribution function represented by the Greek letter Φ (phi). This is typically the area under the distribution curve from negative infinity to x.
$\mathcal{N}(x\|\mu,\sigma^2)$	Normal distribution with mean μ and standard deviation of σ. The variance is σ^2. This distribution is also known as a "Gaussian distribution."
μ	The mean, also known as the expected value of a distribution. It's represented by the Greek letter μ (mu).
σ	The standard deviation of a probability distribution. This gives an idea for how far apart samples are spread apart. For this reason, it's also referred to as the "spread." It's represented by the Greek letter σ (sigma).
$\|\boldsymbol{\Sigma}\|$	The determinant of the $\boldsymbol{\Sigma}$ matrix**.**
$\boldsymbol{\Sigma}^{-1}$	The inverse of the $\boldsymbol{\Sigma}$ matrix.
$\boldsymbol{\Sigma}^{\mathbf{T}}$	The transpose of the $\boldsymbol{\Sigma}$ matrix.
$\int_a^b f(x)dx$	The integral of a function between "a" and "b." I find it helpful to think of an integral as a generic way of multiplying. The presence of this in this paper proves that calculus is actually useful in real life ☺
$\exp(\mathrm{x})$	The exponential function e^x . It is the basic rate of growth for things that grow continuously.
$P(X)$	The marginal probability that "X" will occur. This is also known as the "probability mass function" for discrete events (ones you can count). For continuous values, we call it the "probability density function." We use an uppercase "P" when X is discrete and a lowercase "p" when "X" is continuous.
$P(E\|F)$	The conditional probability of the event "E" occurring given that "F" has occurred.
$P(E \cap F)$	The probability of that the events "E" and "F" will both occur.
$\mathbb{I}(t_2 = p_2 + p_3)$	The "$\mathbb{I}$" represents the Indicator Function. You can effectively ignore this detail and just focus on the bit inside the braces.

Bernoulli Trials

I wrote about flipping a coin in the blog post as a means of building up to probability distributions. I was technically referring to Bernoulli trials leading to a binomial distribution. An outcome in a Bernoulli trial can be a success or a failure. Conceptually, if you did an infinite number of trials, you'll arrive at a Gaussian distribution.[1]

Because a fair coin is expected to have a 50% chance of getting either side, we arbitrarily pick heads to be a "success" and tails to be a "failure." The probability of getting "k" heads in "n" trials is given by this formula:

$$\binom{n}{k} p^k (1-p)^{n-k}$$

where

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

is the binomial coefficient that is read as "the number of ways of choosing 'k' items from a population of size 'n'" or simply "n choose k." Additionally, where p = 0.5 to indicate a 50% chance of heads.
The last important thing is that the variance of a binomial distribution is:

$$np(1-p)$$

This implies that the standard deviation is:

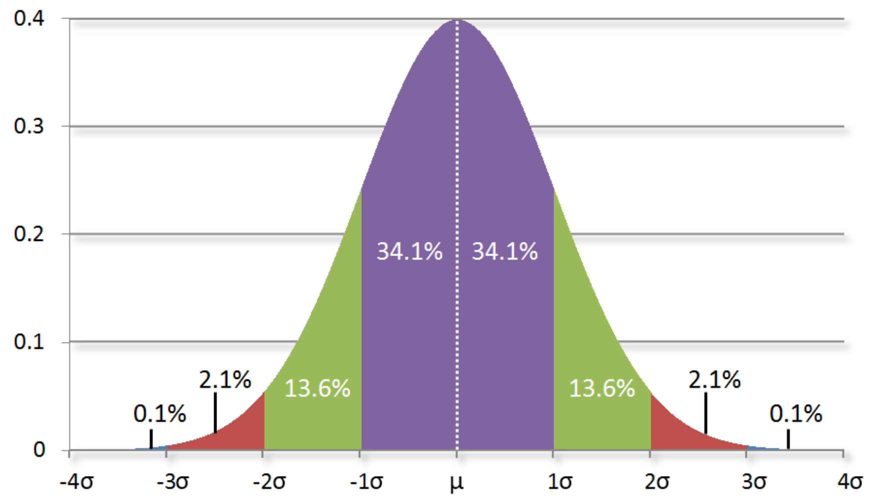
$$\sqrt{np(1-p)}$$

In the post, I implied without proof that the standard deviation of a taking the count of heads after 1000 flips was about 16. This was derived as:

$$\sqrt{np(1-p)} = \sqrt{1{,}000 \cdot 0.5(1-0.5)} = \sqrt{250} \approx 15.81 \approx 16$$

[1] I say "conceptually" because this is roughly what you'd see. It's important to realize that Gaussians are continuous whereas my bar chart histograms showing outcomes were not because they had spaces between each discrete sample. The rough idea is there though if you imagine that the gap between each sample shrinks to zero.

Gaussian Distribution (a.k.a. “Normal Distribution” or “Bell Curve”)



For a single dimension, the value of the normal distribution curve at a given point (e.g. the probability density) is given by this equation:

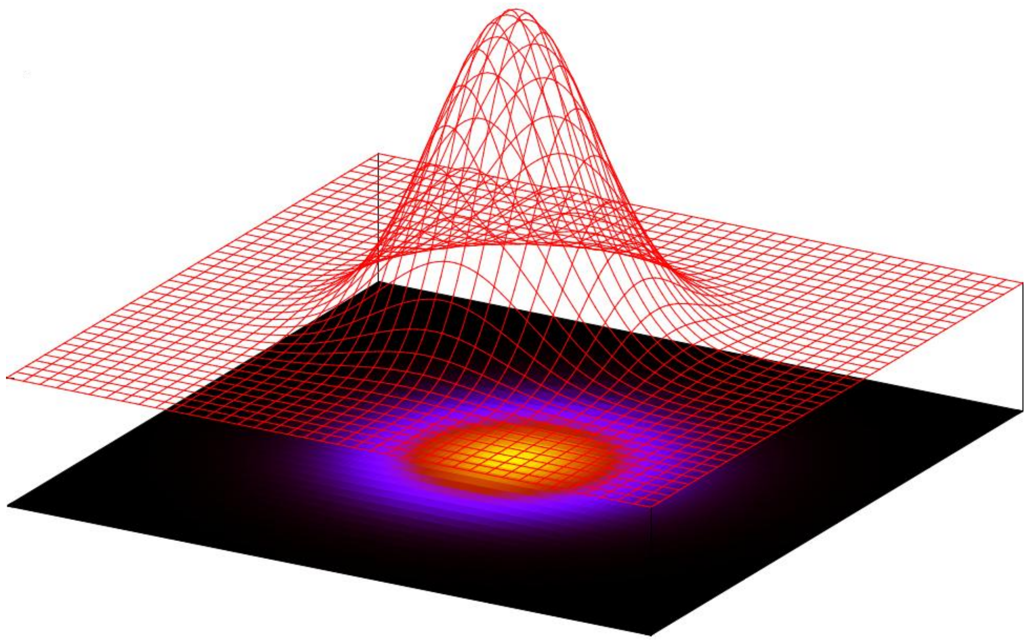
$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

And in higher dimensions:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \cdot \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Here, **Σ** is a matrix whose diagonal values are the variances and D is the number of dimensions. Notice that if D is one, you get the simplified equation above.

As mentioned in the post, here’s an example of a Gaussian in higher dimensions (D=2 in this case):



Note that the color of the 2D plot below it indicates the taller parts of the plot, indicating stronger probabilities.

For the curious, I created the 3D image using GNU Plot with the following commands:

```
set pm3d at b
set ticslevel 0.4
set isosample 40,40
splot 1*exp(-(.1*(x-0)*(x-0) + 2 * 0*(x-0)*(y-0) + .1*(y-0)*(y-0)))
```

Standard Normal Distribution

One interesting observation from the Wikipedia page is:

[A]ny other normal distribution can be regarded as a version of the standard normal distribution that has been stretched horizontally by a factor σ and then translated rightward by a distance μ. Thus, μ specifies the position of the bell curve's central peak, and σ specifies the "width" of the bell curve.

Representing a Gaussian Using Precision and Precision Adjusted Mean

As mentioned on page 5 of the TrueSkill paper (1), it's sometimes more convenient to represent a Gaussian by the "precision" and the "precision adjusted mean."

Precision

Precision is just the inverse of the variance. It is represented by the Greek letter π, which is somewhat unfortunate because it could be confused with the math constant that is approximately 3.14. Specifically:

$$\pi := \sigma^{-2} = \frac{1}{\sigma^2}$$

Precision Adjusted Mean

The precision adjusted mean is simply the precision multiplied by the mean. It is represented by the Greek letter τ. Specifically:

$$\tau := \pi\mu$$

Example

To see why it's convenient to use precision, let's look at multiplication of Gaussians using both methods.

First, from the Multiplying Gaussians section in the Appendix: Fun Stuff with Gaussians, we find:

$$\mathcal{N}(\mu_f, \sigma_f) \cdot \mathcal{N}(\mu_g, \sigma_g) = \mathcal{N}\left(\frac{\mu_f \sigma_g^2 + \mu_g \sigma_f^2}{\sigma_g^2 + \sigma_f^2}, \sqrt{\frac{\sigma_f^2 \sigma_g^2}{\sigma_g^2 + \sigma_f^2}}\right)$$

Using precision, this is simply:

$$\mathcal{N}(\pi_f, \tau_f) \cdot \mathcal{N}(\pi_g, \tau_g) = \mathcal{N}(\pi_f + \pi_g, \tau_f + \tau_g)$$

As you can see, it's an impressive simplification and it's the reason why this representation is used in the code. Just to prove it's valid, we can verify it quickly.

First, let's verify the precision:

$$\pi_f + \pi_g = \frac{1}{{\sigma_f}^2} + \frac{1}{{\sigma_g}^2} = \frac{1}{\sigma_{fg}^2}$$

In the Appendix: Fun Stuff with Gaussians, we derive this:

$$\sigma_{fg} = \sqrt{\frac{\sigma_f^2 \sigma_g^2}{\sigma_g^2 + \sigma_f^2}}$$

If we substitute this in, we'll obtain:

$$\frac{1}{{\sigma_f}^2} + \frac{1}{{\sigma_g}^2} = \frac{1}{\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2 + \sigma_f^2}} = \frac{\sigma_g^2 + \sigma_f^2}{\sigma_f^2\sigma_g^2}$$

To prove that this is valid, simply multiply both sides by $\sigma_f^2\sigma_g^2$:

$$(\sigma_f^2\sigma_g^2)\left(\frac{1}{{\sigma_f}^2} + \frac{1}{{\sigma_g}^2}\right) = (\sigma_f^2\sigma_g^2)\left(\frac{\sigma_g^2 + \sigma_f^2}{\sigma_f^2\sigma_g^2}\right)$$

$$\frac{\cancel{\sigma_f^2}\sigma_g^2}{\cancel{{\sigma_f}^2}} + \frac{\sigma_f^2\cancel{\sigma_g^2}}{\cancel{{\sigma_g}^2}} = (\cancel{\sigma_f^2\sigma_g^2})\left(\frac{\sigma_g^2 + \sigma_f^2}{\cancel{\sigma_f^2\sigma_g^2}}\right)$$

$$\sigma_g^2 + \sigma_f^2 = \sigma_g^2 + \sigma_f^2$$

… and we see that we have proven the equality.

Precision adjusted mean is a little harder:

$$\tau_f + \tau_g = \frac{\mu_f}{{\sigma_f}^2} + \frac{\mu_g}{{\sigma_g}^2} = \frac{\mu_{fg}}{\sigma_{fg}^2}$$

Substituting both the mean a variance values we proved in the Appendix: Fun Stuff with Gaussians:

$$\frac{\mu_f}{{\sigma_f}^2} + \frac{\mu_g}{{\sigma_g}^2} = \frac{\frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\sigma_g^2 + \sigma_f^2}}{\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2 + \sigma_f^2}} = \frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\cancel{\sigma_g^2 + \sigma_f^2}} \cdot \frac{\cancel{\sigma_g^2 + \sigma_f^2}}{\sigma_f^2\sigma_g^2} = \frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\sigma_f^2\sigma_g^2}$$

Multiplying both sides:

$$(\sigma_f^2\sigma_g^2)\left(\frac{\mu_f}{{\sigma_f}^2} + \frac{\mu_g}{{\sigma_g}^2}\right) = (\cancel{\sigma_f^2\sigma_g^2})\left(\frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\cancel{\sigma_f^2\sigma_g^2}}\right)$$

$$\frac{\cancel{\sigma_f^2}\sigma_g^2\mu_f}{\cancel{{\sigma_f}^2}} + \frac{\sigma_f^2\cancel{\sigma_g^2}\mu_g}{\cancel{{\sigma_g}^2}} = \mu_f\sigma_g^2 + \mu_g\sigma_f^2$$

$$\sigma_g^2\mu_f + \sigma_f^2\mu_g = \mu_f\sigma_g^2 + \mu_g\sigma_f^2$$

And once again, we've proven equality. It's amazing how simple multiplication and division are using this little substitution trick.

For completeness:

$$\frac{\mathcal{N}(\pi_f, \tau_f)}{\mathcal{N}(\pi_g, \tau_g)} = \mathcal{N}(\pi_f - \pi_g, \tau_f - \tau_g)$$

The proof is similar to the one above.

Partial Update

A partial update is when we only apply a percentage of the full update. This is achieved by representing a player's skill in terms of precision and precision adjusted mean that we defined earlier.

Let's say that the TrueSkill algorithm tells us that a player's new full skill update should be:

$$\mathcal{N}(\pi_{\text{new}}, \tau_{\text{new}})$$

Now, let's assume that instead of a "full" update, we just want a "partial" update. We can pick some value between 0% and 100% to denote how much of an update we want. We'll call that percentage "a."

Now we define the partial update function:

$$PartialUpdate(\mathcal{N}(\pi_{\text{new}}, \tau_{\text{new}}), a) = \mathcal{N}(\pi_{\text{old}} + a(\pi_{\text{new}} - \pi_{\text{old}}), \tau_{\text{old}} + a(\tau_{\text{new}} - \tau_{\text{old}}))$$

Thus, a partial update adds only a percentage of the full update.

Using Maxima to do the grunt work, we can transform this back into the normal form of a Gaussian using mean and standard deviation:

$$\sigma_{\text{after partial update}} = \frac{1}{\sqrt{a\left(\frac{1}{\sigma_{\text{new}}{}^2} - \frac{1}{\sigma_{\text{old}}{}^2}\right) + \frac{1}{\sigma_{\text{old}}{}^2}}} = \frac{\sigma_{\text{old}}\sigma_{\text{new}}}{\sqrt{a\sigma_{\text{old}}{}^2 - (a-1)\sigma_{\text{new}}{}^2}}$$

And

$$\mu_p = \frac{(a\mu_{\text{old}} - \mu_{\text{old}})\sigma_{\text{new}}{}^2 - \mu_{\text{new}}a\sigma_{\text{old}}{}^2}{(a-1)\sigma_{\text{new}}{}^2 - a\sigma_{\text{old}}{}^2} = \frac{\mu_{\text{old}}(a-1)\sigma_{\text{new}}{}^2 - \mu_{\text{new}}a\sigma_{\text{old}}{}^2}{(a-1)\sigma_{\text{new}}{}^2 - a\sigma_{\text{old}}{}^2}$$

The equations in traditional form look much more complicated, but if you look closely, you can sort of see how the percentage multiplier affects the outcome. If nothing else, look how the 0% and 100% cases simplify.

Paired Comparisons

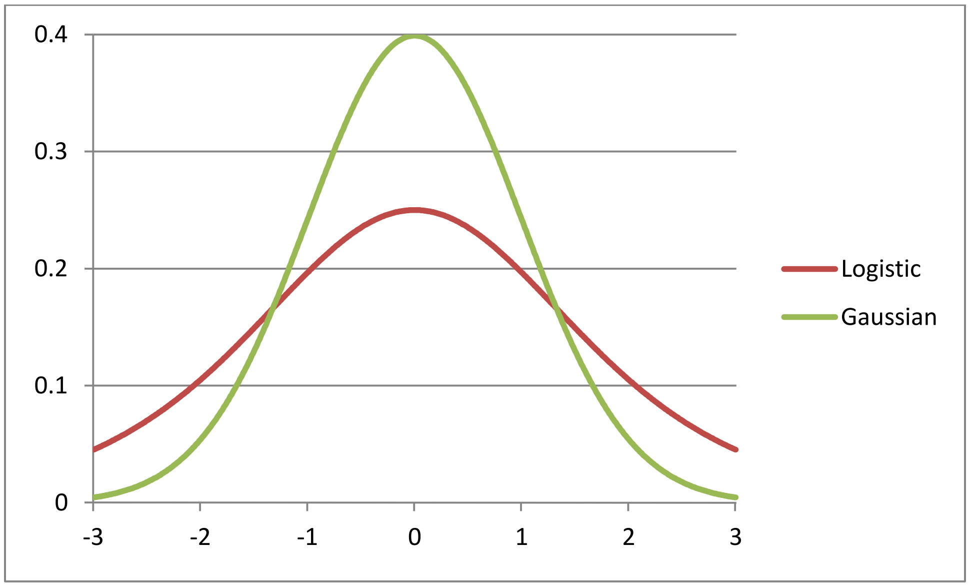
Page 1 of the TrueSkill paper (1) briefly mentions Thurstone Case V and Bradley-Terry pairwise comparisons. I won't go into details here either, but it's interesting to research this, especially Thurstone Case V and how it was developed in the 1920s in the context of how children compare the severity of crimes.

Elo Curves

Arpad Elo originally used a Gaussian distribution, but the chess folks found that a logistic curve fits real data better (and it's easier to program as well since Gaussian functions aren't built into most standard math libraries).

As you can see in the accompanying source code, the concepts are identical; it's just that a different curve is used.

The difference can be seen:



Elo Skill Update

The first page of the TrueSkill (1) paper shows the Elo equations of:

$$P(p_1 > p_2 | s_1, s_2) = \Phi\left(\frac{s_1 - s_2}{\sqrt{2}\beta}\right)$$

Leading to an update equation of:

$$\Delta = \alpha\beta\sqrt{\pi}\left(\frac{y+1}{2} - \Phi\left(\frac{s_1 - s_2}{\sqrt{2}\beta}\right)\right)$$

It seems clear that the $s_1 - s_2$ part comes from the fact that we're dealing with the subtraction of two Gaussian curves (Adding and Subtracting Gaussians) that have the same standard deviation which will lead to a combined standard deviation of $\sqrt{2}\beta$. By dividng out the $\sqrt{2}\beta$, we get a standard normal leading to a traditional cumulative distribution function. Effectively, it's telling us how many standard deviations away from the mean.

If you're curious about the seemingly obscure "$\sqrt{2}\beta$" bit, remember that it came from the subtraction convolution of two Gaussians that have the same β standard deviation:

$$\sigma_{f\otimes g} = \sqrt{{\sigma_f}^2 + {\sigma_g}^2}$$

$$= \sqrt{\beta^2 + \beta^2}$$

$$= \sqrt{2\beta^2}$$

$$= \sqrt{2}\beta$$

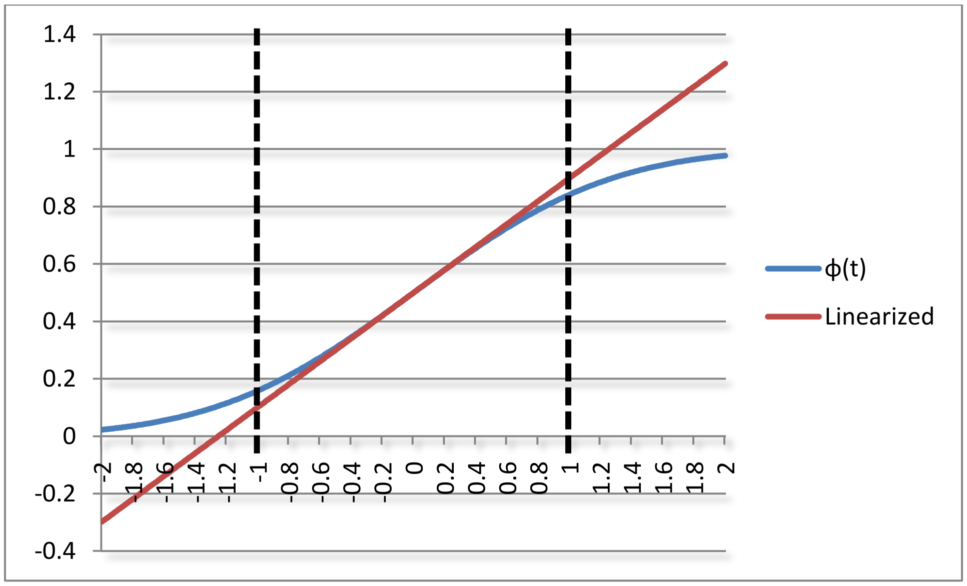
See the Adding and Subtracting Gaussians section in the appendix for more details.

K-Factor

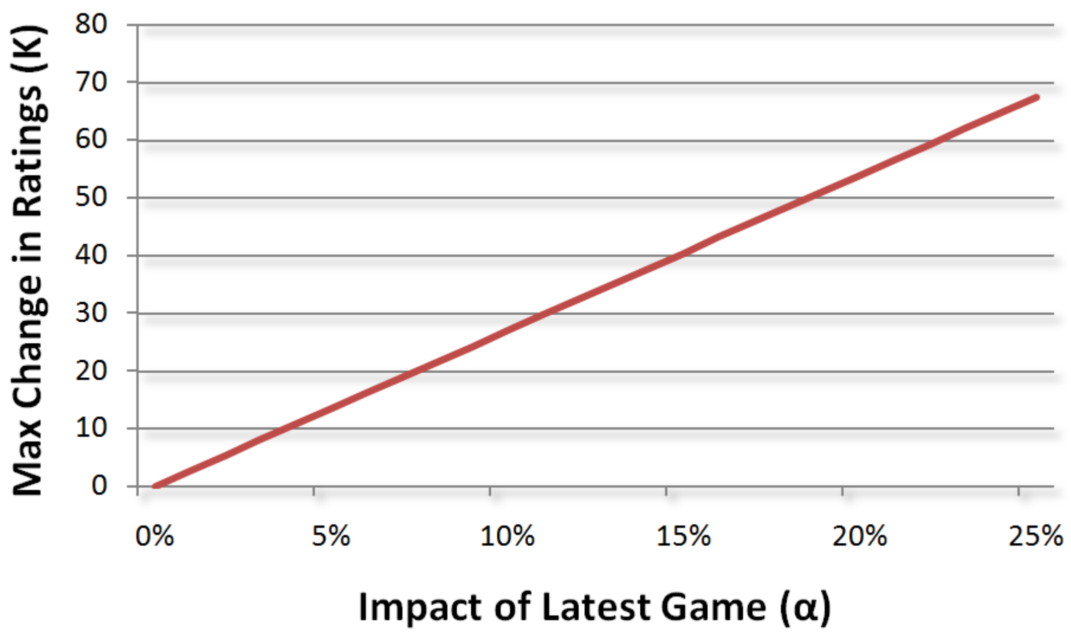
The most curious part I found about the Elo update equation is the presence of the $\sqrt{\pi}$. This comes from approximating the cumulative distribution function in the region of +-1 standard deviation by a straight line:

$$\Phi(t) \approx \frac{1}{2} + \frac{t}{\sqrt{2\pi}}, \qquad -1 < t < 1$$

The reasoning is that you shouldn't play someone beyond a standard deviation away from you, so the linearized approximation is ok. You can visually see that this is a reasonable linear approximation under these conditions:



As mentioned in the post, here's how the K-factor updates as the alpha value changes from 0% to 25%:



Note that typical values for α are between 5% and 10% leading to a K-factor of approximately 10 to 30. The higher your chess ranking, the less likely you want to risk it fluctuating much. For this reason, games with grandmasters typically have a smaller α and therefore a smaller K-factor.

Elo Example

In the post, I used an example of playing a beginner. Here's how the values were updated.

My new score:

$$= 1200 + \Delta$$

$$= 1200 + 24 \cdot \left(\frac{-1+1}{2} - \Phi\left(\frac{1200 - 1000}{\sqrt{2}\beta} \right) \right)$$

$$= 1200 + 24 \cdot \left(\frac{0}{2} - \Phi\left(\frac{200}{\sqrt{2} \cdot 200} \right) \right)$$

$$= 1200 + 24 \cdot \left(-\Phi\left(\frac{1}{\sqrt{2}\,\cdot} \right) \right)$$

$$= 1200 - 24 \cdot \Phi\left(\frac{1}{\sqrt{2}\,\cdot} \right)$$

$$\approx 1200 - 18.246$$

$$\approx 1181.754$$

$$\approx 1182$$

Likewise, the beginner's rating would now be:

$$= 1000 + \Delta$$

$$= 1000 + 24 \cdot \left(\frac{1+1}{2} - \Phi\left(\frac{1000 - 1200}{\sqrt{2}\beta} \right) \right)$$

$$= 1000 + 24 \cdot \left(\frac{2}{2} - \Phi\left(\frac{-200}{\sqrt{2} \cdot 200} \right) \right)$$

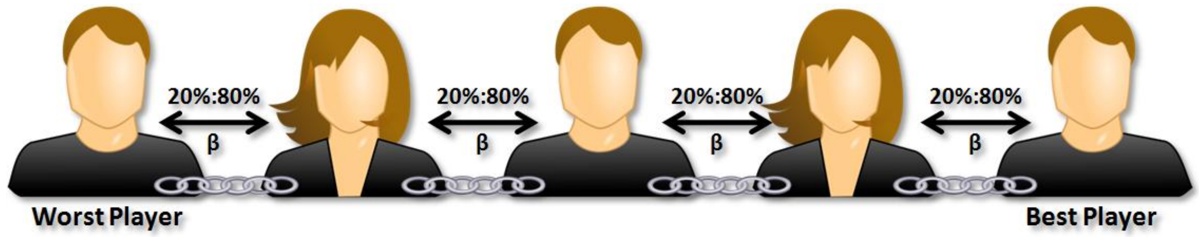
$$= 1000 + 24 \cdot \left(1 - \Phi\left(\frac{-1}{\sqrt{2}} \right) \right)$$

$$\approx 1000 + 18.246$$

$$\approx 1018.246$$

$$\approx 1018$$

Beta (β): The Skill Class Width



In (2), TrueSkill co-inventor Ralf Herbrich gives a good definition of β as defining the length of the "skill chain." If a game has a wide range of skills, then β will tell you how wide each link is in the skill chain. This can also be thought of how wide (in terms of skill points) each skill class.

Similiarly, β tells us the number of skill points a person must have above someone else to identify an 80% probability of win against that person.

For example, if β is 4 then a player Alice with a skill of "30" will tend to win against Bob who has a skill of "26" approximately 80% of the time.

Tau (τ): The Additive Dynamics Factor

Without τ, the TrueSkill algorithm would always cause the player's standard deviation (σ) term to shrink and therefore become more certain about a player. Before skill updates are calculated, we add in τ^2 to the player's skill variance (σ^2). This ensures that the game retains "dynamics." That is, the τ parameter determines how easy it will be for a player to move up and down a leaderboard. A larger τ will tend to cause more volatility of player positions.

TrueSkill Default Values

As mentioned on page 8 of the TrueSkill paper (1), the initial values for a player are:

$$\mu_0 = 25, \sigma_0^2 = \left(\frac{25}{3}\right)^2$$

This leads to an initial TrueSkill ($\mu - 3\sigma$) of zero.

The default values for a game are:

$$\beta^2 = \left(\frac{\sigma_0}{2}\right)^2, \tau^2 = \left(\frac{\sigma_0}{100}\right)^2$$

This leads to reasonable dynamics, but you might need to adjust as needed.

Calculating a Leaderboard

One of the most important aspects of ranking is displaying a leaderboard. As mentioned on page 8 of the TrueSkill paper (1), one way of doing this is to compute the conservative skill estimate for each player (the TrueSkill) of $\mu - 3\sigma$ then sort by that.

Draw Margin

The draw margin is discussed on page 6 of the TrueSkill paper (1). We see it listed as

$$\text{draw probability} = \Phi\left(\frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right) - \Phi\left(\frac{-\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right) = 2\Phi\left(\frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right) - 1$$

Since the rest of the TrueSkill equations require that we know the draw margin (ε) we'll solve for it in terms of the draw probability (P_{draw}) and the inverse cumulative distribution function (Φ^{-1}):

$$P_{draw} = 2\Phi\left(\frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right) - 1$$
$$P_{draw} + 1 = 2\Phi\left(\frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right)$$
$$\frac{P_{draw} + 1}{2} = \Phi\left(\frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}\right)$$
$$\Phi^{-1}\left(\frac{P_{draw} + 1}{2}\right) = \frac{\varepsilon}{\sqrt{n_1 + n_2}\,\beta}$$
$$\varepsilon = \Phi^{-1}\left(\frac{P_{draw} + 1}{2}\right)\sqrt{n_1 + n_2}\,\beta$$

This equation is used inside the source code for computing the draw margin from a game's draw probability.

Bayesian Probability

Bayesian probability begins with the definition of conditional probability:

$$P(E \cap F) = P(E|F)P(F)$$

This means that the probability of both "E" and "F" occurring is the probability of "E" given that "F" has occurred multiplied by the probability of "F" occurring. This makes intuitive sense.

Note that we could have just as easily written this as:

$$P(E \cap F) = P(F|E)P(E)$$

Setting these two equal, we get:

$$P(E|F)P(F) = P(E \cap F) = P(F|E)P(E)$$

$$P(E|F)P(F) = P(F|E)P(E)$$

Rearranging terms we get:

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}$$

This is known as Bayes formula. The author of (3) puts it this way:

$$\text{posterior} = \frac{\text{likelihood } \times \text{prior}}{\text{evidence}}$$

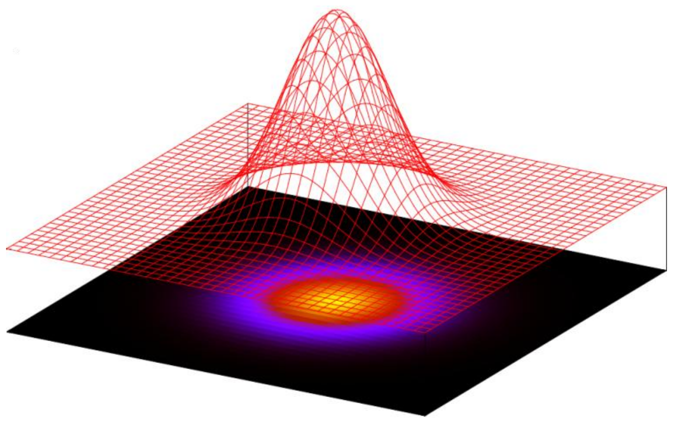
That is, our new belief (the posterior) of the probability of "E" given that we've observed "F" is the product of the likelihood of observing "F" given that "E" has been observed multiplied by our prior belief of "E" occurring. In order to normalize things (e.g. make sure everything sums to 1), we divide out by the "evidence" which is the probability of "F" occurring, regardless of what we've observed.

The fundamental idea is that the Bayesian approach multiplies likelihood by a prior to obtain a posterior.

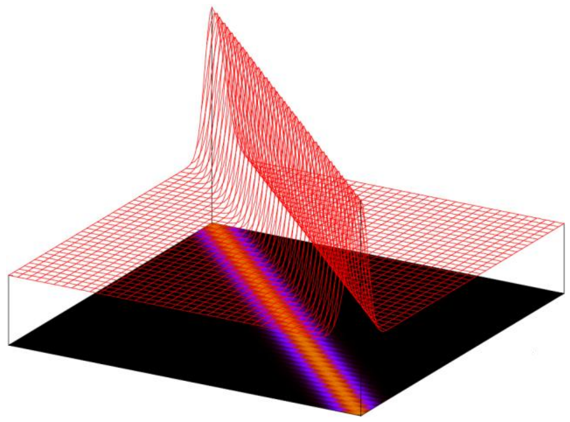
Visual Explanation: 3D

Here is an example of what things look like in 3D:

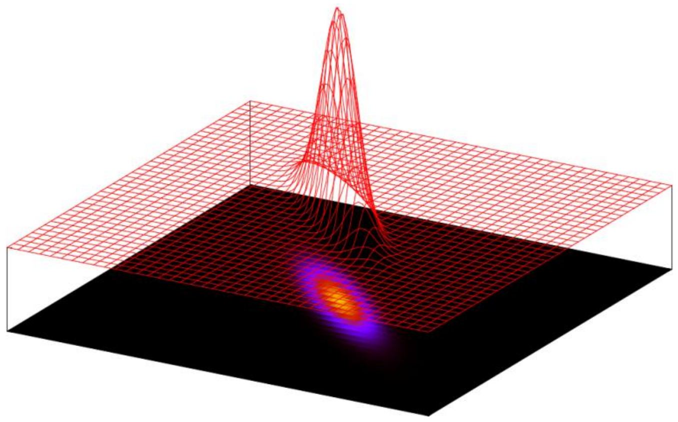
Prior



Likelihood



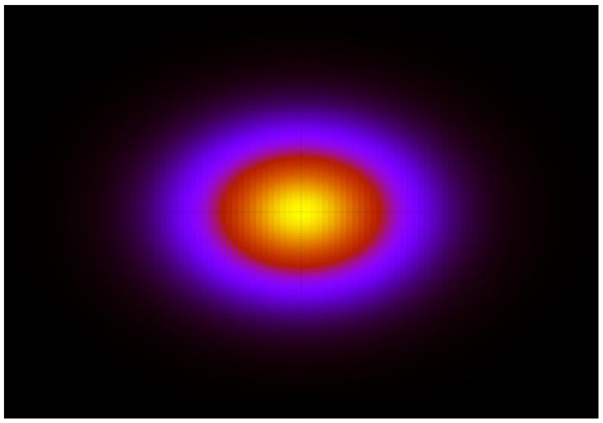
Posterior



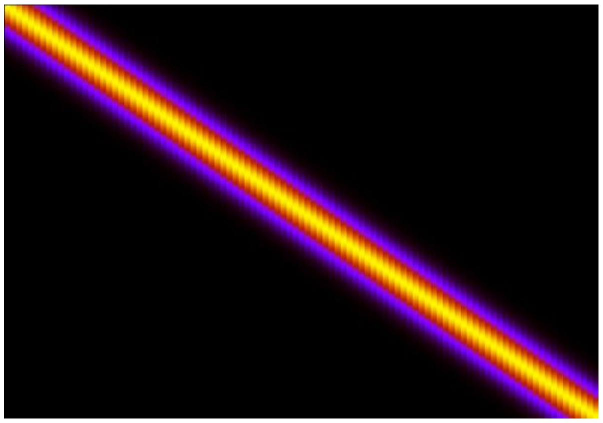
Visual Explanation: 2D

Here is the same example in 2D:

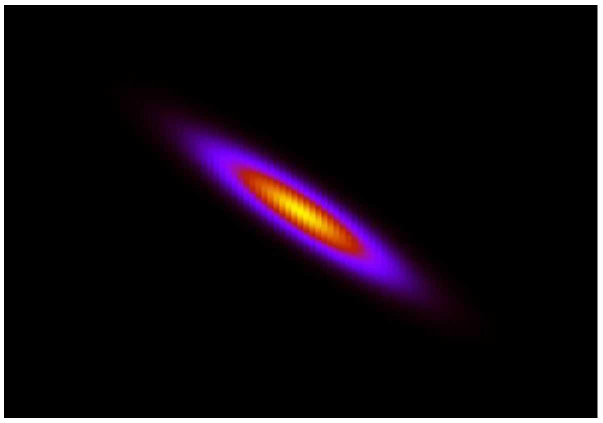
Prior



Likelihood:



Posterior



Bayesian Example 1: Probability of Cancer

What is the probability that you actually have breast cancer (C_+) given that your mammogram test indicates you have cancer (T_+)? That is, we want to know $P(C_+|T_+)$.

Let’s borrow from (4) and assume you have the following *prior* information on breast cancer:

- 1% of women have breast cancer ($P(C_+) = 0.01$)
    - This implies that 99% of women *do not* have cancer ($P(C_-) = 0.99$)
- Mammograms detect cancer 80% of the time when it is actually present ($P(T_+|C_+) = 0.80$)
    - This implies that mammograms *do not* indicate cancer 20% of the time when it is present: $P(T_-|C_+) = 0.20$.
- 9.6% of mammograms falsely report that you have cancer when you actually do not have it ($P(T_+|C_-) = 0.096$).
    - This implies that there is a 90.4% chance of correctly reporting that you don’t have cancer when you indeed do not have cancer ($P(T_-|C_-) = 0.904$).

Bayes’ formula gives us:

$$P(C_+|T_+) = \frac{P(T_+|C_+) \cdot P(C_+)}{P(T_+|C_+) \cdot P(C_+) + P(T_+|C_-) \cdot P(C_-)}$$

Notice how this is the *likelihood* of having a positive cancer test result given that you actually have cancer multiplied by the *prior* probability of having cancer. In addition, we divide by the *evidence* which is the probability of getting *any* positive test result.

Using the actual values gives us:

$$P(C_+|T_+) = \frac{0.80 \cdot 0.01}{0.80 \cdot 0.01 + 0.096 \cdot 0.99}$$

$$= \frac{0.008}{0.008 + 0.09504}$$

$$= \frac{0.008}{0.10304}$$

$$\approx .0776$$

$$\approx 7.76\%$$

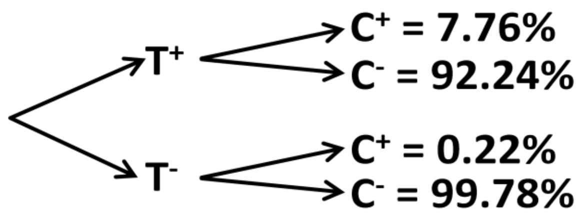
Given the relatively high false-positive rate of this test, we only have a 7.76% chance of actually having cancer if we have a positive test result. This is somewhat non-intuitive.

For more details on this, see (4).

Using a Bayesian Decision Tree

Another way of looking at the above example is to represent it as a tree where each branching point represents a decision that could go several ways.

For example:



The far left of the tree represents the two possibilities of test outcomes and the next decision represents if you have cancer. Note that the top outcome is what we calculated earlier.

Bayesian Example 2: Probability of Spam

What is the probability that an email is spam given the words of the email?

Let's say you get an email that starts off like this:

First I must solicit your confidence in this transaction. This is by virtue of its nature as being utterly confidential and top secret. We are top officials of the Federal Government Contract Review Panel who are interested in importation of goods into our country with funds which are presently trapped in Nigeria. In order to commence this business we solicit your assistance to enable us RECIEVE the said trapped funds ABROAD.

A computer can look at this email and break it up into words and then calculate the probability of it being spam using Bayes' formula:

$$P(\text{spam}|\text{word}) = \frac{P(\text{word}|\text{spam}) \cdot P(\text{spam})}{P(\text{word}|\text{spam}) \cdot P(\text{spam}) + P(\text{word}|\text{ham}) \cdot P(\text{ham})}$$

Note that in this equation, I refer to "ham" as the equivalent of "not spam."

From *prior* experience, we know that 90% of our email is spam, so $P(\text{spam}) = 0.90$. Also, in the past we've had users train our filter by classifying email as either "spam" or "ham." From this training, we have *likelihoods* for $P(\text{word}|\text{spam})$ and $P(\text{word}|\text{ham})$.

Additionally, we'll use a simpler algorithm called "Naïve Bayes" because we're going to make a naïve assumption that words in this email are independent events. This is definitely not true for English. For example, "Federal Government" is much more likely than a phrase like "Federal utterly," but we'll find out that it doesn't matter much for classification purposes.

Therefore, the determination of "spaminess" of this particular email can be done like this:

$$P(\text{spam}|\text{words}) = \frac{P(\text{spam}|\text{first}) \cdot P(\text{spam}|\text{i}) \cdots P(\text{spam}|\text{nigeria}) \cdots P(\text{spam}|\text{abroad})}{P(\text{spam}|\text{first}) \cdots P(\text{spam}|\text{abroad}) + P(\text{ham}|\text{first}) \cdots P(\text{ham}|\text{abroad})}$$

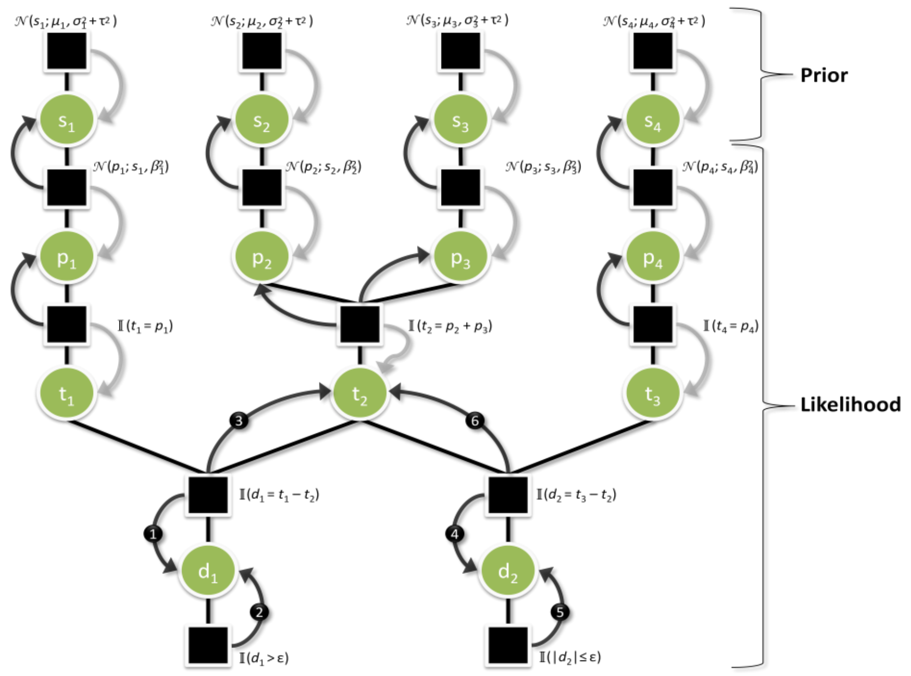
The last major step is to define some cutoff probability where anything above this value is classified as spam. The cutoff should be high enough to limit the chance of false positives (e.g. your friend that always forwards you spammy-like emails) and low enough to actually remove most spam.

Each time that the system is wrong, you can improve its accuracy by specifically marking something as spam and thus adding more data.

(Note that we ignore the casing of the words so that “NIGERIA=Nigeria=Nigeria.” Fancier filters might ignore suffixes of words too, but the idea is the same.)

Factor Graphs

As mentioned in the post, here’s an example of a TrueSkill factor graph:


$\mathcal{N}(s_1;\mu_1,\sigma_1^2+\tau^2)$
$\mathcal{N}(s_2;\mu_2,\sigma_2^2+\tau^2)$
$\mathcal{N}(s_3;\mu_3,\sigma_3^2+\tau^2)$
$\mathcal{N}(s_4;\mu_4,\sigma_4^2+\tau^2)$
Prior
s_1
s_2
s_3
s_4
$\mathcal{N}(p_1;s_1,\beta_1^2)$
$\mathcal{N}(p_2;s_2,\beta_2^2)$
$\mathcal{N}(p_3;s_3,\beta_3^2)$
$\mathcal{N}(p_4;s_4,\beta_4^2)$
p_1
p_2
p_3
p_4
$\mathbb{I}(t_1=p_1)$
$\mathbb{I}(t_2=p_2+p_3)$
$\mathbb{I}(t_4=p_4)$
t_1
t_2
t_3
Likelihood
3
6
$\mathbb{I}(d_1=t_1-t_2)$
$\mathbb{I}(d_2=t_3-t_2)$
1
4
d_1
d_2
2
5
$\mathbb{I}(d_1>\varepsilon)$
$\mathbb{I}(|d_2|\leq\varepsilon)$

Factor graphs are described in detail in (5), but the basic idea is that a factor graph breaks up a joint probability distribution into a graph with two types of nodes (called “bipartite”) of factors (boxes) and variables (circles). The joint distribution is the product of the factors:

$$p(X) = \frac{1}{Z}\prod_{S} f_S(X_S)$$

where X_S represents the variables connected to a factor f_S and Z is a normalization constant.

The core idea of factor graphs is the Sum-Product Update Rule:

The message sent from a node v *on an edge* e *is the product of the local function at* v *(or the unit function if* v *is a variable node) with all messages received at v on edges* other *than e, summarized for the variable associated with* e. *(5)*

The TrueSkill paper (1) shows the important three equations related to factor graphs:

$$p(v_k) = \prod_{f \in F_{v_k}} m_{f \to v_k}(v_k)$$

This tells us that the value of the marginal at v_k is just the product of the incoming messages

$$m_{f \to v_j(v_j)} = \int \cdots \int f(\mathbf{v}) \prod_{i \neq j} m_{v_i \to f}(v_i) d\mathbf{v}_{\setminus j}$$

This indicates that the value of a message from a factor to a variable is the sum of the product of all other messages except the one we're trying to compute along with the value of the function for all v's except for the jth item.

$$m_{v_k \to f}(v_k) = \prod_{\bar{f} \in F_{v_k} \setminus \{f\}} m_{\bar{f} \to k}(v_k)$$

This last equation tells us the message going from variable v_k to the factor f is a product of the other messages.

In (5), these equations are provided as the "variable to local function" message:

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

where $n(x)$ represents the neighbors of the x variable. In addition, there is the "local function to variable" equation:

$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

Again, many more details are provided in (5) along with examples.

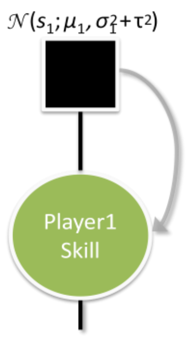
Scheduling on a Factor Graph

In the accompanying source code, there are several classes devoted to "scheduling" things on a factor graph. The "schedule" is just a means of making sure messages are passing between factors.

Now, we need to investigate each factor in the TrueSkill factor graph. We'll do that next.

Gaussian Prior Factors

The prior factor was represented in the post as the black box in this picture:



The prior factor is the easiest to understand. Remember our discussion on precision and precision adjusted mean:

$$\pi := \sigma^{-2} = \frac{1}{\sigma^2}$$

$$\tau := \pi\mu = \frac{\mu}{\sigma^2}$$

The goal of the prior factor is to take the Gaussian represented by a mean "m" and variance "v^2" and update the values to reflect that:

$$\pi_x^{\text{new}} \leftarrow \pi_x + \frac{1}{v^2}$$

$$\tau_x^{\text{new}} \leftarrow \tau_x + \frac{m}{v^2}$$

These match the update equations given in the TrueSkill paper (1).

Factorizing Gaussian Prior

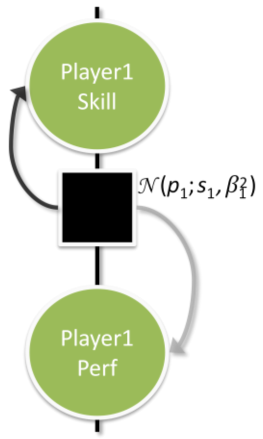
Page 2 of the TrueSkill paper (1) mentions the assumption of a "factorising Gaussian prior distribution" of:

$$P(s) := \prod_{i=1}^{n} \mathcal{N}\left(s_i;\ \mu_i, \sigma_i^2\right)$$

This makes sense because the factor graph itself is one big multiplication to obtain a joint distribution. In this case, the factorizing Gaussian is a multiplication of these prior factors.

Gaussian Likelihood Factors

The Gaussian Likelihood factor was presented in the post as the black box in:



The basic idea is that we have an existing Gaussian variable y and then we want to come up with a new Gaussian x that incorporates the c^2uncertainty: $\mathcal{N}(x; y, c^2)$

We're told:

$$a := \left(1 + c^2(\pi_y - \pi_{f\to y})\right)^{-1} = \frac{1}{1 + c^2(\pi_y - \pi_{f\to y})}$$

$$\approx \frac{1}{1 + c^2\left(\frac{1}{\sigma_y^2}\right)} = \frac{1}{1 + \frac{c^2}{\sigma_y^2}} = \frac{1}{\frac{\sigma_y^2}{\sigma_y^2} + \frac{c^2}{\sigma_y^2}} = \frac{1}{\frac{\sigma_y^2 + c^2}{\sigma_y^2}} = \frac{\sigma_y^2}{\sigma_y^2 + c^2}$$

It appears that a is a multiplier that will always be less than 1. It adds in the c^2 variance appropriately:

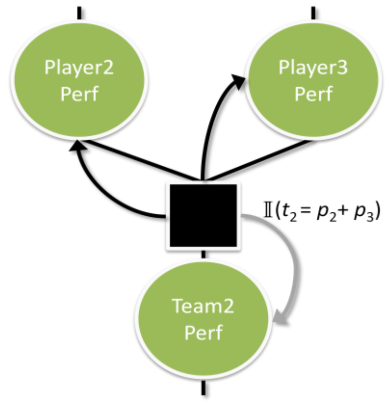
$$\pi_{f\to x}^{\text{new}} \leftarrow a(\pi_y - \pi_{f\to y}) \approx \frac{\sigma_y^2}{\sigma_y^2 + c^2} \cdot \frac{1}{\sigma_y^2} = \frac{1}{\sigma_y^2 + c^2}$$

Likewise, for precision adjusted mean:

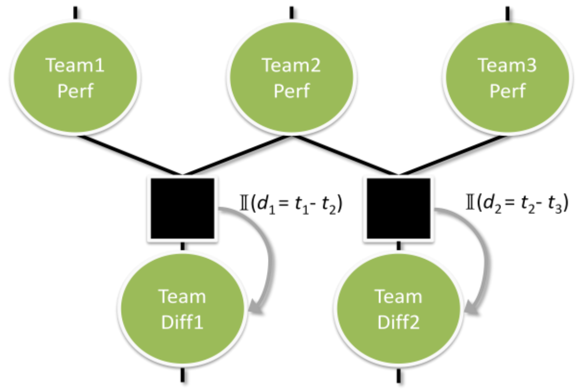
$$\tau_{f\to x}^{\text{new}} \leftarrow a(\tau_y - \tau_{f\to y}) \approx \frac{\sigma_y^2}{\sigma_y^2 + c^2} \cdot \frac{\mu_y}{\sigma_y^2} = \frac{\mu_y}{\sigma_y^2 + c^2}$$

Gaussian Weighted Sum Factors

The Gaussian Weighted Sum factor was presented in the post as the black box in:



In addition, this factor was used for team differences:



This factor takes a bunch of Gaussians and then sums them together. Let's assume that the sum variable is v_0. We'll also assume that the variables we want to sum are $v_1, v_2, \cdots, v_{n-1}$. In addition, we'll assume each sum is weighted by a factor a_n. This means we can write the sum as:

$$v_0 = a_1 v_1 + a_2 v_2 + \cdots + a_{n-1} v_{n-1} + a_n v_n$$

In the appendix we cover Adding and Subtracting Gaussians in detail, but here we'll just use that result and extend it to sum an arbitrary number of Gaussians. This is called the "Normal Sum Distribution" and is covered in (6). The key result is that the sum of Gaussians is also Gaussian with a mean:

$$\mu = \sum_{i=1}^{n} a_i \mu_i$$

and whose variance is:

$$\sigma^2 = \sum_{i=1}^{n} a_i^2 \sigma_i^2$$

Remember how earlier we covered how it is sometimes easier to work with precision (π) and precision mean (τ)? We'll need to convert the above results into that format:

$$\pi^{\text{new}} = \frac{1}{\sigma^2} = \frac{1}{\sum_{j=1}^{n} a_j^2 \sigma_j^2} = \frac{1}{\sum_{j=1}^{n} a_j^2 \frac{1}{\pi_j}} = \frac{1}{\sum_{j=1}^{n} \frac{a_j^2}{\pi_j}} = \left(\sum_{j=1}^{n} \frac{a_j^2}{\pi_j}\right)^{-1}$$

The precision mean will multiply the above precision by the mean:

$$\tau^{\text{new}} = \pi^{\text{new}} \cdot \mu = \pi^{\text{new}} \cdot \left(\sum_{j=1}^{n} a_j \mu_j\right)$$

However, we can rewrite mean as the precision mean divided by the precision:

$$\mu = \frac{\tau}{\pi}$$

This will give us:

$$\tau^{\text{new}} = \pi^{\text{new}} \cdot \left(\sum_{j=1}^{n} a_j \mu_j\right) = \pi^{\text{new}} \cdot \left(\sum_{j=1}^{n} a_j \frac{\tau_j}{\pi_j}\right)$$

This gives a hint at the weighted sum update equations given in the TrueSkill paper.

Now, let's solve for v_1 by subtracting from both sides:

$$v_0 - a_1 v_1 = a_2 v_2 + \cdots + a_{n-1} v_{n-1} + a_n v_n$$

Then, we subtract out v_0:

$$-a_1 v_1 = a_2 v_2 + \cdots + a_{n-1} v_{n-1} + a_n v_n - v_0$$

Now, we just divide by $-a_1$:

$$v_1 = \frac{a_2}{-a_1} v_2 + \cdots + \frac{a_{n-1}}{-a_1} v_{n-1} + \frac{a_n}{-a_1} v_n - \frac{1}{-a_1} v_0$$

Now, we can simplify this by factoring out the negative value:

$$v_1 = -\frac{a_2}{a_1} v_2 - \cdots - \frac{a_{n-1}}{a_1} v_{n-1} - \frac{a_n}{a_1} v_n + \frac{1}{a_1} v_0$$

Similarly, we can solve for the other variables like v_2:

$$v_0 - a_2 v_2 = a_1 v_1 + \cdots + a_{n-1} v_{n-1} + a_n v_n$$

$$-a_2 v_2 = a_1 v_1 + \cdots + a_{n-1} v_{n-1} + a_n v_n - v_0$$

$$v_2 = -\frac{a_1}{a_2} v_1 - \cdots - \frac{a_{n-1}}{a_2} v_{n-1} - \frac{a_n}{a_2} v_n + \frac{1}{a_2} v_0$$

… and v_n:

$$v_0 - a_n v_n = a_1 v_1 + a_2 v_2 + \cdots + a_{n-1} v_{n-1}$$

$$-a_n v_n = a_1 v_1 + a_2 v_2 + \cdots + a_{n-1} v_{n-1} - v_0$$

$$v_n = -\frac{a_1}{a_n} v_1 - \frac{a_2}{a_n} v_2 - \cdots - \frac{a_{n-1}}{a_n} v_{n-1} + \frac{1}{a_n} v_0$$

Note that we can simplify this by factoring out the $\frac{1}{a_n}$ that is common to all terms:

$$v_n = \frac{1}{a_n}(-a_1 v_1 - a_2 v_2 - \cdots - a_{n-1} v_{n-1} + 1 v_0)$$

We can also rewrite this in vector notation:

$$y_n = \mathbf{a}^{\mathrm{T}}[y_1, y_2, \cdots, y_{n-1}, x]$$

Where:

$$a = \frac{1}{b_n} \cdot \begin{bmatrix} -b_1 \\ -b_2 \\ \vdots \\ -b_{n-1} \\ +1 \end{bmatrix}$$

It is the above compact notation that we see in the TrueSkill paper (1).

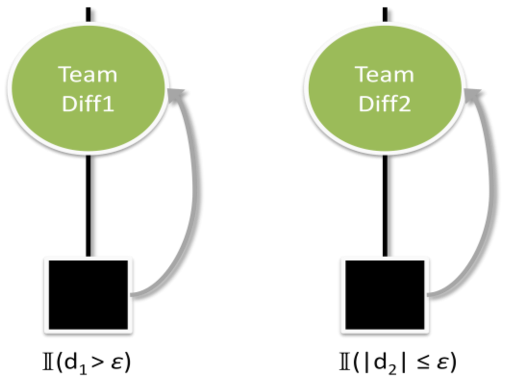
Partial Play

In (2), Ralf Herbrich discusses how TrueSkill supports the concept of “partial play.” This allows the algorithm to properly handle situations where a player wasn’t present for the entire duration of the game. This is implemented in the team performance Gaussian weighted sum factor where the weights are the percentage of the time a player was present.

The accompanying source code supports partial play. Refer to that for more details.

Gaussian Comparison Factors

The comparison factors were presented in the post in both their non-drawn and drawn versions:


Team Diff1
Team Diff2
$\mathbb{I}(d_1 > \varepsilon)$
$\mathbb{I}(|d_2| \leq \varepsilon)$

The bottom of the TrueSkill factor graph consists of comparison factors depending on the actual observed outcome of the game. These functions make use of “v” and “w” functions that I refer to as TruncatedGaussianCorrectionFunctions in the code.

The full details of these functions are described in (7), but the basic idea is that you have a three dimensional Gaussian created by the performances of the two teams. Visually, it looks like this:

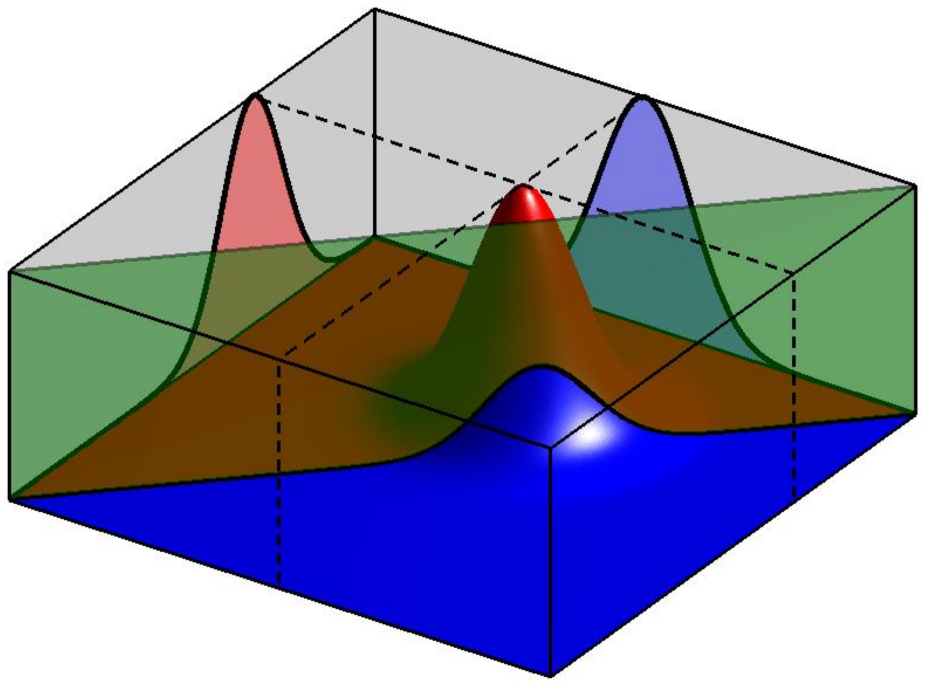


Figure 1 TrueSkill Gaussians image by Ralf Herbrich and Thore Graepel, Microsoft Research Cambridge.

In the above picture, the red team is playing the blue team. If the red team wins, we “chop” the 3D Gaussian by setting the blue losing team portion to all zeros. This process would leave us with a “truncated Gaussian” that we approximate with another Gaussian using a technique called Expectation Propagation. This is sort of hand-wavy description, but the full details are described in (7).

Even though this is probably the most complicated part of the algorithm, you can still get an idea for what's happening by looking closely at the simplified two-player equations in (8).

We'll cover each one in the next two sections.

Mean Additive Truncated Gaussian Function: "v"

The first function is "v" and it helps to determine how much to update the mean after a win or loss:

$$\mu_{winner} \longleftarrow \mu_{winner} + \frac{\sigma_{winner}^2}{c} \cdot v\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\varepsilon}{c}\right)$$

$$\mu_{loser} \longleftarrow \mu_{loser} - \frac{\sigma_{loser}^2}{c} \cdot v\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\varepsilon}{c}\right)$$

Because we add to a player's existing mean, we'll refer to it as the "additive" factor.

Note that in all these equations, we have a normalizing "c" value:

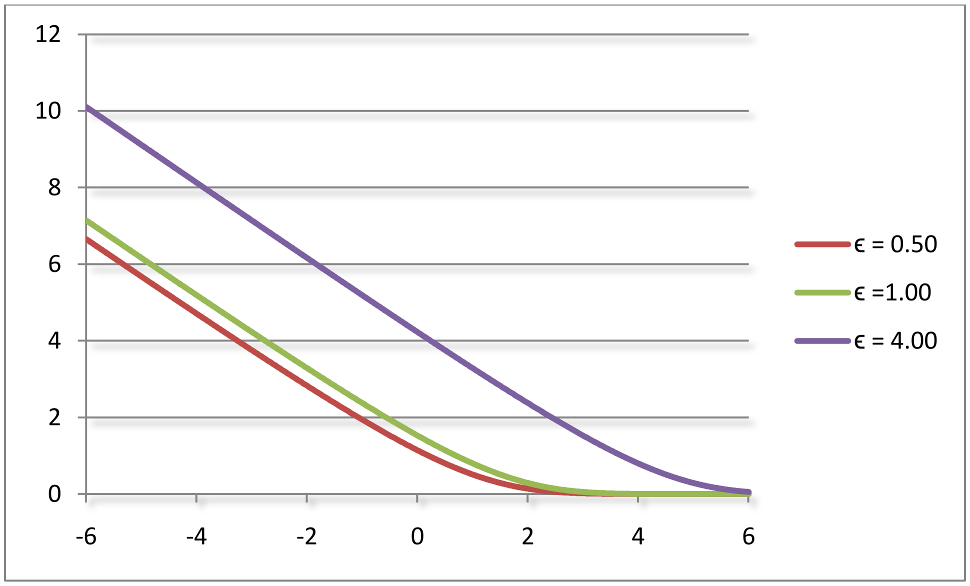
$$c^2 = 2\beta^2 + \sigma_{winner}^2 + \sigma_{loser}^2$$

Non-draw

In the TrueSkill paper, we're told that the non-draw version of "v" has this equation:

$$V_{\mathbb{I}(\cdot > \varepsilon)} := \frac{\mathcal{N}(t-\varepsilon)}{\Phi(t-\varepsilon)}$$

The best way to get a feel for this and others is to look at a plot of it with respect to "t":

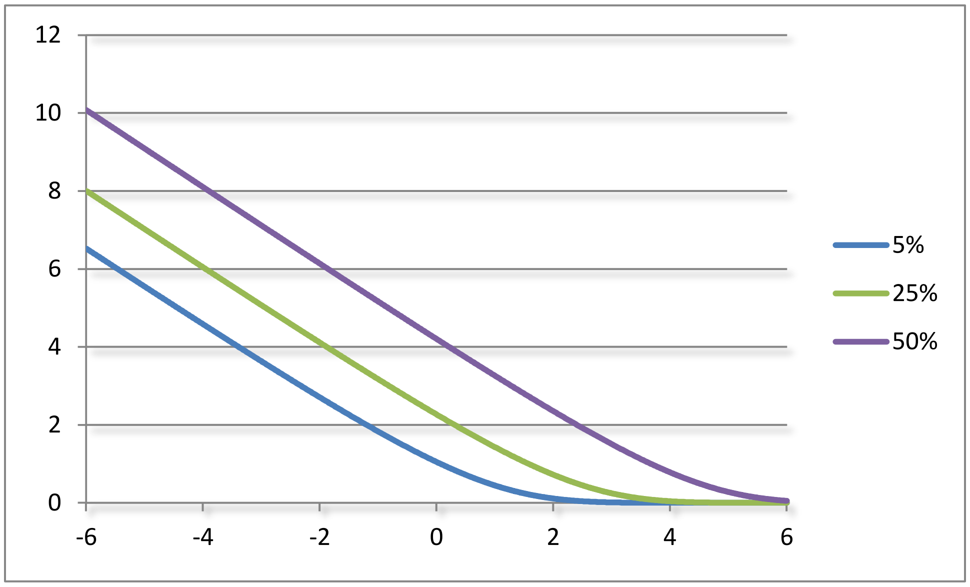


Here, you can see that if "t" is negative, there is a larger update. A negative "t" indicates an "upset" victory meaning that the expected person didn't win. Again, looking at

$$t = \frac{(\mu_{winner} - \mu_{loser})}{c}$$

shows us that a negative "t" implies that the winner's mean was *less* than the loser's mean. Before the match, the system expects a non-negative "t" based on what it had already learned. A positive "t" value indicates an expected outcome and thus little need for an update (as reflected in the graph).

In all of these functions, the ε value indicates the "draw margin." This is a non-intuitive way to think about it. If we use the equation we derived earlier, we can calculate the corresponding draw probability for a given ε and then use that instead. Here's a plot that shows how the draw probability affects "v":



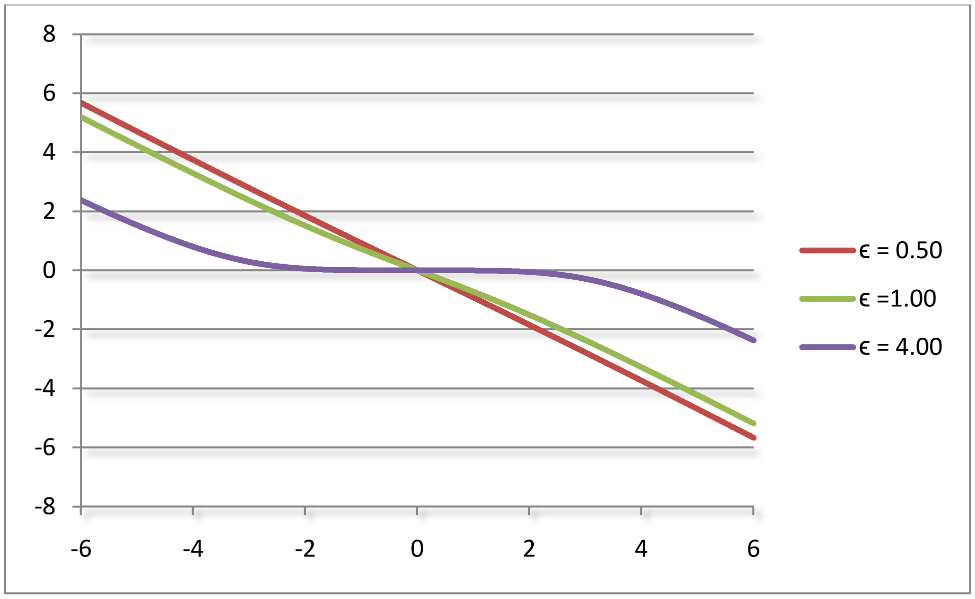
Note that larger draw probabilities lead to larger updates for this non-drawn case. Another way to think about this is that we've observed a win. If we had higher draw probability, it would have been less likely to actually observe a win, so we'll have a larger update because it was unexpected.

Draw Version

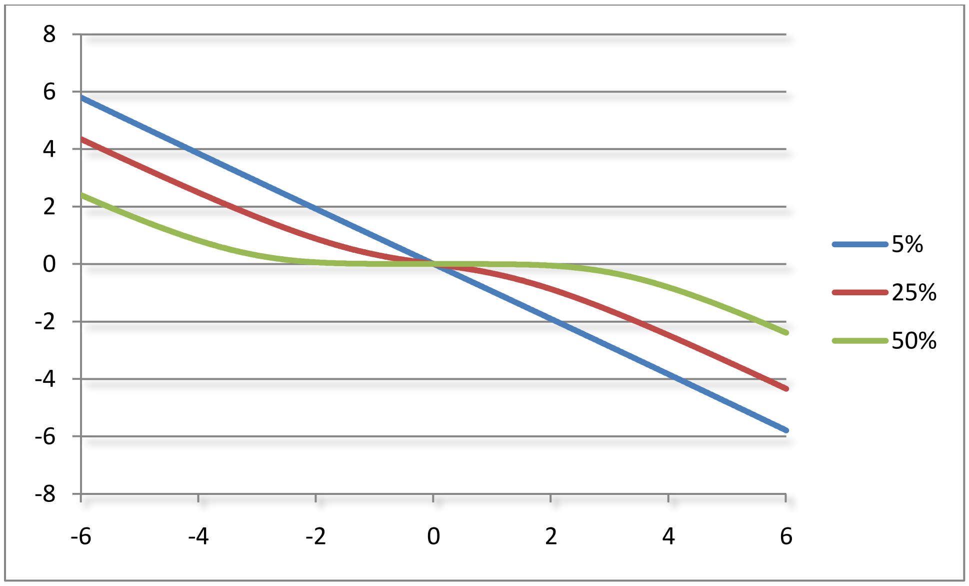
TrueSkill explicitly models draws. This means that we have separate update equations for draws. In particular, the draw version of "v" is:

$$V_{\mathbb{I}(|\cdot| \leq \varepsilon)} := \frac{\mathcal{N}(-\varepsilon - t) - \mathcal{N}(\varepsilon - t)}{\Phi(\varepsilon - t) - \Phi(-\varepsilon - t)}$$

It has a corresponding plot of:



We can easily convert this to draw probabilities using what we already know:



As you can see, if “t” is negative, then we have an “upset” where we were expecting the better player to win, but they ended up having a draw against a worse player. If the probability of a draw is low, the update is bigger because the probability of actually observing a draw was low.

Variance Multiplicative Function: “w”

If we look at the equations given in (8) for updating the standard deviation, we see:

$$\sigma_{winner}^2 \leftarrow \sigma_{winner}^2 \cdot \left[1 - \frac{\sigma_{winner}^2}{c^2} \cdot w\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\varepsilon}{c}\right)\right]$$

$$\sigma^2_{loser} \leftarrow \sigma^2_{loser} \cdot \left[1 - \frac{\sigma^2_{loser}}{c^2} \cdot w\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\varepsilon}{c}\right)\right]$$

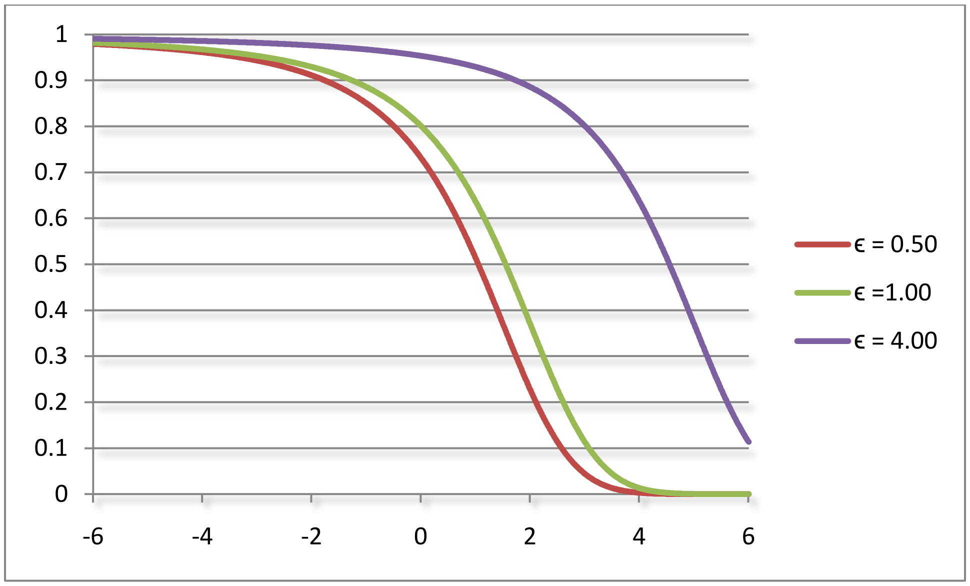
Because we multiply the variance by "w", we can consider "w" to be a multiplicative factor. Like "v", it has a drawn and non-drawn version:

Non-drawn

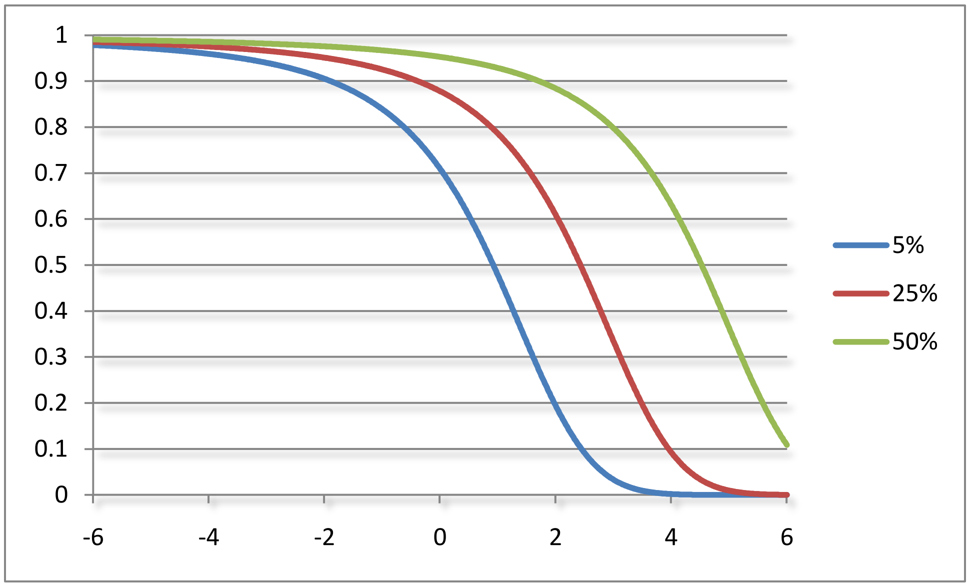
The non-drawn version is defined as:

$$W_{\mathbb{I}(\cdot>\varepsilon)}(t,\varepsilon) := V_{\mathbb{I}(\cdot>\varepsilon)}(t,\varepsilon) \cdot \left(V_{\mathbb{I}(\cdot>\varepsilon)}(t,\varepsilon) + t - \varepsilon\right)$$

It has this plot:



Converting to draw probabilities gives us:



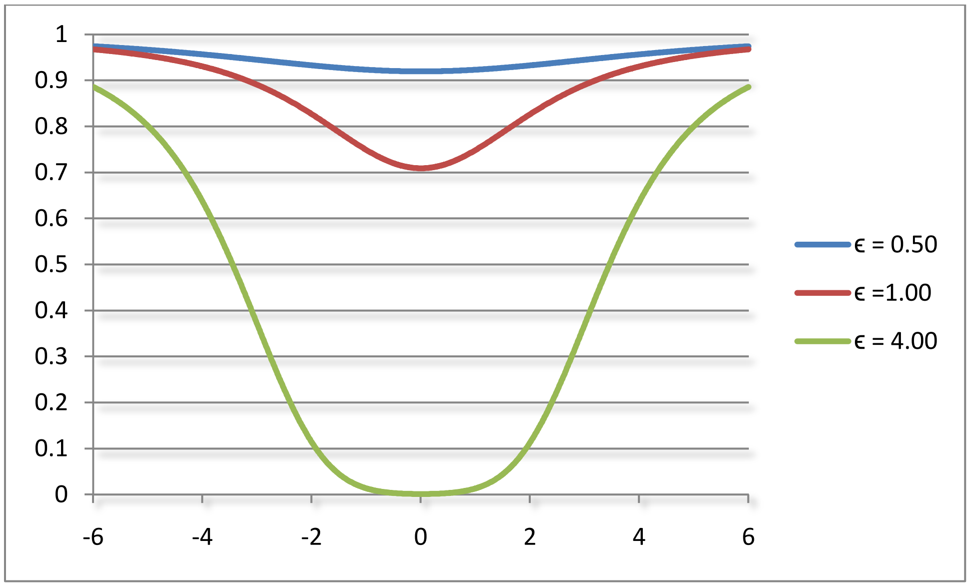
This tells that that we should reduce our estimate of a player's standard deviation if we observe a big upset (e.g. a much stronger player *losing* to a weaker player). Additionally, the more likely a draw was to happen, the more we should reduce the standard deviation (e.g. the uncertainty) because it was an unexpected event.

Drawn

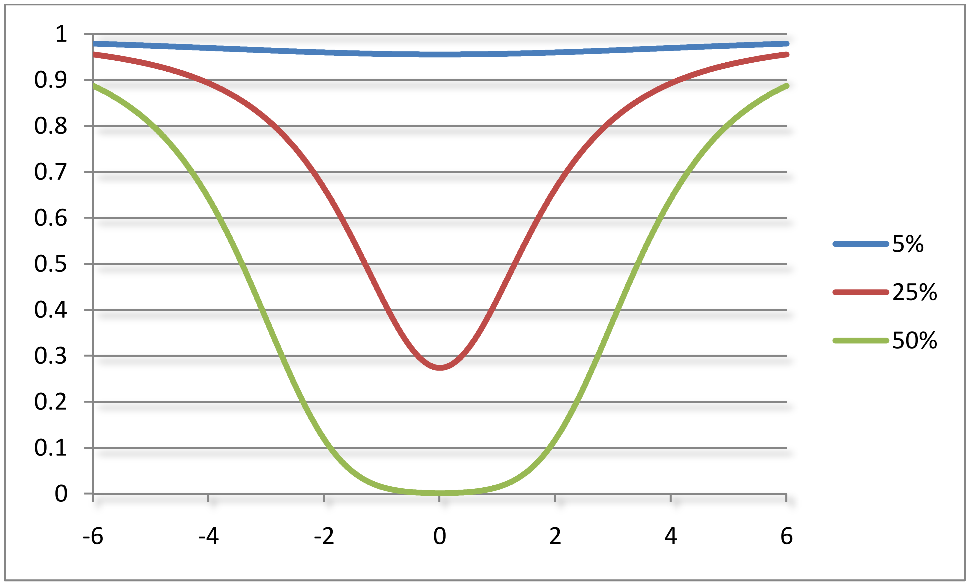
The drawn version of "w" is the most complicated:

$$W_{\mathbb{I}(|\cdot|\leq\varepsilon)}(t,\varepsilon) := V^2_{\mathbb{I}(|\cdot|\leq\varepsilon)}(t,\varepsilon) + \frac{(\varepsilon - t)\cdot\mathcal{N}(\varepsilon - t) + (\varepsilon + t)\cdot\mathcal{N}(\varepsilon + t)}{\Phi(\varepsilon - t) - \Phi(-\varepsilon - t)}$$

It has a nice symmetrical plot:



Again, converting to draw probabilities:



Here we can see that the smallest update occurs when players are expected to draw. Otherwise, it's symmetric with respect to the difference.

Match Quality

On page 8 of the TrueSkill paper (1), we read:

Pairwise matchmaking of players is performed using a match quality criterion derived as the draw probability relative to the highest possible draw probability in the limit $\varepsilon \to 0$.

I was confused after reading this and wrote the authors for clarity. Ralf Herbrich was kind to offer additional information. In particular, the paper refers to this limit:

$$P(draw|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \lim_{\varepsilon \to 0} \int_{-\varepsilon 1}^{+\varepsilon 1} \mathcal{N}(\mathbf{z}; \boldsymbol{A}^T \boldsymbol{\mu}; \boldsymbol{A}^T (\beta^2 \boldsymbol{I} + \boldsymbol{\Sigma}) \boldsymbol{A})\, d\mathbf{z}$$

This integral is effectively summing up the Gaussian probability density function in the region of the draw margin.

Symbol	Meaning
$\boldsymbol{\mu}$	A vector of all of the skill means
$\boldsymbol{\Sigma}$	A matrix whose diagonal values are the variances (σ^2) of each of the players.
β	The game's beta factor mentioned earlier in this paper.
$\boldsymbol{A}$	The player-team assignment and comparison matrix

The "A" Matrix

The $\boldsymbol{A}$ matrix requires some further explanation since it combines team assignments *and* comparisons. It is a matrix whose rows represent players and whose columns represent team comparisons. This matrix is also aware of partial play (mentioned earlier). In addition, it's important to realize that since team comparisons are made, you'll always have one less column than teams since "n" teams have "n-1" successive pairwise comparisons.

An example can go a long way. Consider a 3 team game where team 1 just consists of player 1, team 2 is player 2 and player 3, and team 3 is just player 4. Furthermore, player 2 and player 3 on team 2 played 25% and 75% of the time respectively (e.g. partial play), the $\boldsymbol{A}$ matrix for this situation is:

$$\begin{bmatrix} +1.00 & 0 \\ -0.25 & +0.25 \\ -0.75 & +0.75 \\ 0 & -1.00 \end{bmatrix}$$

Note how we have positive values for the current team and negative values for the next team in the comparison.

Match Quality Further Derivation

With further insight from Ralf, I was able to follow the derivation. We take the limit mentioned above and divide it by a normalizing value indicating a perfect match (e.g. all the teams have the same team skill, that is $\boldsymbol{A}^T\boldsymbol{\mu} = \boldsymbol{0}$):

$$q_{\text{draw}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta, \boldsymbol{A}) := \frac{\mathcal{N}(\boldsymbol{0}; \boldsymbol{A}^T\boldsymbol{\mu}; \boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A})}{\mathcal{N}(\boldsymbol{0}; \boldsymbol{0};\ \beta^2\boldsymbol{A}^T\boldsymbol{A})}$$

We can calculate this value using the multivariate Gaussian equation we saw earlier of:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \cdot \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

We can now substitute the values:

$$q_{\text{draw}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \beta, \boldsymbol{A}) := \frac{\frac{1}{(2\pi)^{\frac{D}{2}}} \cdot \frac{1}{\sqrt{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{0}-\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(\boldsymbol{0}-\boldsymbol{A}^T\boldsymbol{\mu})}}{\frac{1}{(2\pi)^{\frac{D}{2}}} \cdot \frac{1}{\sqrt{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{0}-\boldsymbol{0})^{\mathrm{T}}(\beta^2\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{0}-\boldsymbol{0})}}$$

$$= \frac{\frac{1}{\sqrt{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{0}-\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(\boldsymbol{0}-\boldsymbol{A}^T\boldsymbol{\mu})}}{\frac{1}{\sqrt{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}} e^{\boldsymbol{0}}}$$

$$= \frac{\sqrt{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}}{\sqrt{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(-\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(-\boldsymbol{A}^T\boldsymbol{\mu})}$$

$$= \sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(-1)(\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(-1)(\boldsymbol{A}^T\boldsymbol{\mu})}$$

$$= \sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(-1)(-1)(\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(\boldsymbol{A}^T\boldsymbol{\mu})}$$

$$= \sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{A}^T\boldsymbol{\mu})^{\mathrm{T}}\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(\boldsymbol{A}^T\boldsymbol{\mu})}$$

$$= \sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{A})\left(\boldsymbol{A}^T(\beta^2\boldsymbol{I}+\boldsymbol{\Sigma})\boldsymbol{A}\right)^{-1}(\boldsymbol{A}^T\boldsymbol{\mu})}$$

$$= \sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\boldsymbol{A}^T(\beta^2\boldsymbol{I} + \boldsymbol{\Sigma})\boldsymbol{A}|}} e^{-\frac{1}{2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{A})\left(\beta^2\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}\right)^{-1}(\boldsymbol{A}^T\boldsymbol{\mu})}$$

And from here you can reduce it to the equation that Ralf gave me:

$$\exp\left(-\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{A}(\beta^2\boldsymbol{A}^T\boldsymbol{A}+\ \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\mu}\right)\cdot\sqrt{\frac{|\beta^2\boldsymbol{A}^T\boldsymbol{A}|}{|\beta^2\boldsymbol{A}^T\boldsymbol{A}+\ \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}|}}$$

It's interesting to see what happens in the simple case of two players. In this case, we have:

$$\boldsymbol{\mu}=\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix},\boldsymbol{\Sigma}=\begin{bmatrix}\sigma_i^2 & 0\\ 0 & \sigma_j^2\end{bmatrix},\boldsymbol{A}=\begin{bmatrix}1\\ -1\end{bmatrix}$$

If we plug these into the equation we just derived, we get:

$$=\exp\left(-\frac{1}{2}\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}\left(\beta^2\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}+\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\sigma_i^2 & 0\\ 0 & \sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right)^{-1}\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}\right)$$
$$\cdot\sqrt{\frac{\left|\beta^2\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}\right|}{\left|\beta^2\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}+\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\sigma_i^2 & 0\\ 0 & \sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right|}}$$

We remember from our matrix math days that:

$$\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}=\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}=1^2+\ (-1)^2=2$$

This leads to the simplification:

$$=\exp\left(-\frac{1}{2}\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}^T\begin{bmatrix}1\\ -1\end{bmatrix}\left(2\beta^2+\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\sigma_i^2 & 0\\ 0 & \sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right)^{-1}\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}\right)$$
$$\cdot\sqrt{\frac{|2\beta^2|}{\left|2\beta^2+\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\sigma_i^2 & 0\\ 0 & \sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right|}}$$

Using normal matrix-multiplication, we can simplify further:

$$=\exp\left(-\frac{1}{2}(\mu_i-\mu_j)\left(2\beta^2+\ \begin{bmatrix}\sigma_i^2 & -\sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right)^{-1}\begin{bmatrix}1\\ -1\end{bmatrix}^T\begin{bmatrix}\mu_i\\ \mu_j\end{bmatrix}\right)\cdot\sqrt{\frac{|2\beta^2|}{\left|2\beta^2+\ \begin{bmatrix}\sigma_i^2 & -\sigma_j^2\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}\right|}}$$

$$=\exp\left(-\frac{1}{2}(\mu_i-\mu_j)(2\beta^2+\ \sigma_i^2+\ \sigma_j^2)^{-1}(\mu_i-\mu_j)\right)\cdot\sqrt{\frac{|2\beta^2|}{|2\beta^2+\ \sigma_i^2+\ \sigma_j^2|}}$$

And now we're back to normal real numbers (or you might think of them as 1x1 matrices). The inverse and determinant of a 1x1 matrix is simple:

$$a^{-1} = \frac{1}{a}, |a| = a$$

Giving us this simplification:

$$= \exp\left(-\frac{1}{2}\frac{(\mu_i - \mu_j)^2}{2\beta^2 + \sigma_i^2 + 2\sigma_j^2}\right) \cdot \sqrt{\frac{2\beta^2}{2\beta^2 + \sigma_i^2 + \sigma_j^2}}$$

That further simplifies to:

$$= \exp\left(-\frac{(\mu_i - \mu_j)^2}{2(2\beta^2 + \sigma_i^2 + 2\sigma_j^2)}\right) \cdot \sqrt{\frac{2\beta^2}{2\beta^2 + \sigma_i^2 + \sigma_j^2}}$$

Almost surprisingly, all the simplification works and we're left with the match quality equation 4.1 in the TrueSkill paper for two players:

$$q_{\text{draw}}(\beta^2, \mu_i, \mu_j, \sigma_i, \sigma_j) := \sqrt{\frac{2\beta^2}{2\beta^2 + \sigma_i^2 + \sigma_j^2}} \cdot \exp\left(-\frac{(\mu_i - \mu_j)^2}{2(2\beta^2 + \sigma_i^2 + \sigma_j^2)}\right)$$

Had we not looked at the general case, this equation would probably not make as much sense.

Kullback-Leibler Divergence

Page 4 of the TrueSkill paper (1) mentions minimizing Kullback-Leibler divergence (a.k.a. "K-L divergence"). One way to think about this is to think of two probability distributions P and Q. For example, these could be complicated shapes or maybe something as simple as a Gaussian distribution.

Assume that P is the real distribution and that Q is the model that approximates it. The Kullback-Leibler divergence is a distance measure of how much extra information is required to add to a sample from Q (the approximation) to get the actual value from P (the real thing). Thus, when the TrueSkill authors speak of minimizing this metric, it means that their Gaussian approximation (Q) is similar to the real thing (P).

In TrueSkill "P" is the 3D truncated Gaussian we talked about earlier and "Q" is the approximated Gaussian.

Convergence Properties

On page 7 of the TrueSkill paper (1), the authors write:

TrueSkill comes close to the information theoretic limit of $n \log(n)$ bits to encode a ranking of n players. For 8 player games, the information theoretic limit is $\log(n) / \log(8) \approx 5$ games per player on average and the observed convergence for these two players is ≈ 10 games!

This makes some intuitive sense because the simplest way of encoding a ranking is to assign each player a number between 1 and "n" (or technically, "0" and "n-1"). The number of bits required to encode numbers of size "n" is $\log_2(n)$.

What makes less sense is that 8 player games would require:

$$\frac{\log_2(n)}{\log_2(8)} \approx 5 \text{ games}$$

I'm assuming that "n" was known for the Halo game beta described and was somewhere around

$$2^{5 \cdot \log_2(8)} = 2^{5 \cdot 3} = 2^{15} = 32768 \text{ players.}$$

This number seems to coincide with a medium-sized beta group for free-for-all.

Appendix: Fun Stuff with Gaussians

Throughout this paper and in the post, I've hinted at some properties of Gaussians. The derivations are somewhat tedious, so I left them for the end. In this section, I'll expand on them.

Deriving the Gaussian Normalization Constant

The normalization constant for a Gaussian is what we have to divide it by to ensure that the sum of all probabilities (e.g. the integral) is one.

Borrowing from (9), we can see that a Gaussian can be written like this:

$f(x) = e^{-\frac{x^2}{2\sigma^2}}$ where $x \in [-\infty, \infty]$

Note that this is equivalent to

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Because we want the area under the curve, any non-zero value for μ would just shift the graph on the x-axis by μ. Since the graph goes forever on each side, this really makes it not matter much. This means we can ignore the mean in terms calculating the integration constant. Additionally, we can introduce $\lambda = \frac{1}{2\sigma^2}$ to help simplify things.

The area under the curve is not 1, but we can figure it out using an integration trick:

$$I(\lambda) = \int_{-\infty}^{\infty} e^{-\lambda x^2} dx$$

Although somewhat counter-intuitive, we can actually simplify things by taking the square of this:

$$I^2(\lambda) = \left(\int_{-\infty}^{\infty} e^{-\lambda x^2} dx\right)^2$$

$$= \left(\int_{-\infty}^{\infty} e^{-\lambda x^2} dx\right)\left(\int_{-\infty}^{\infty} e^{-\lambda x^2} dx\right)$$

Once again, we can simplify by making it slightly more complicated by introducing another variable. Instead of using the x variable twice, we can make the second integral use y:

$$I^2(\lambda) = \left(\int_{-\infty}^{\infty} e^{-\lambda x^2} dx\right)\left(\int_{-\infty}^{\infty} e^{-\lambda y^2} dy\right)$$

This can be converted a double integral that you'd have learned about in a third-semester calculus class:

$$I^2(\lambda) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\lambda x^2}\, e^{-\lambda y^2} dx\, dy$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\lambda(x^2+y^2)}\, dx\, dy$$

This is effectively integrating across the entire Cartesian plane (the one with x and y axis). We can rethink about this in terms of polar coordinates. This transforms the integral into thinking of it as integrating around a circle. Instead of integrating to plus and minus infinity on each axis, we think of it as a circle with a radius that goes from 0 to infinity and then we go around the entire circle. This can be expressed as:

$$x^2 + y^2 = r^2$$

Where:

$$dxdy = r\, d\theta dr$$

The limits cover the entire circle of the polar coordinate system:

$\theta \in [0, 2\pi], r \in [0, \infty]$

After the transform to polar coordinates, the integral becomes:

$$I^2(\lambda) = \int_0^{2\pi}\int_0^{\infty} re^{-\lambda r^2} dr d\theta$$

The θ isn't used by the inner integral, so we can split it up as:

$$I^2(\lambda) = \int_0^{2\pi} d\theta \int_0^{\infty} re^{-\lambda r^2} dr$$
$$= \left(\theta\big|_0^{2\pi}\right)\left(-\frac{e^{-\lambda r^2}}{2\lambda}\bigg|_0^{\infty}\right)$$
$$= (2\pi - 0)\left(\lim_{r\to\infty} -\frac{e^{-\lambda r^2}}{2\lambda} + \frac{e^0}{2\lambda}\right)$$

$$= 2\pi\left(0 + \frac{1}{2\lambda}\right)$$
$$= 2\pi\,\frac{1}{2\lambda}$$
$$I^2(\lambda) = \frac{\pi}{\lambda}$$

Now, all we have to do is take the square root:

$$I(\lambda) = \sqrt{\frac{\pi}{\lambda}}$$

We recall from earlier that $\lambda = \frac{1}{2\sigma^2}$, this means:

$$I(\lambda) = \sqrt{\frac{\pi}{\lambda}} = \sqrt{\frac{\pi}{\frac{1}{2\sigma^2}}} = \sqrt{2\sigma^2\pi} = \sigma\sqrt{2\pi}$$

Thus, you often see a "normalized" Gaussian distribution where we divide out this constant up front:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

Gaussian Moments

In reading about Gaussians, you'll typically run across the term "moments" and "expectations." The thing to keep in mind is the order of the moment. For example, the first "moment" is the mean:

$$\mathbb{E}[x] = \mathbb{E}[x^1] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu,\sigma^2)\, x^1\, dx = \mu$$

We can go up to higher "moments" by increasing the power of x. For example, the second moment is:

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu,\sigma^2)\, x^2\, dx = \mu^2 + \sigma^2$$

And if you do some symbol manipulation, you can get to variance:

$$var[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = (\mu^2 + \sigma^2) - (\mu)^2 = \sigma^2$$

Other moments can be interesting, but we'll ignore those for now.

Multiplying Gaussians

Multiplying a distribution sounds odd, but it sort of makes sense. We borrow from (10)

$f(x) = \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{\left(x-\mu_f\right)^2}{2\sigma_f^2}}$ and $g(x) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{\left(x-\mu_g\right)^2}{2\sigma_g^2}}$

We can go ahead and just multiply these two directly:

$$f(x)g(x) = \left(\frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{\left(x-\mu_f\right)^2}{2\sigma_f^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{\left(x-\mu_g\right)^2}{2\sigma_g^2}}\right)$$

Collecting terms

$$= \left(\frac{1}{\sqrt{2\pi}\sigma_f} \cdot \frac{1}{\sqrt{2\pi}\sigma_g}\right)\left(e^{-\frac{\left(x-\mu_f\right)^2}{2\sigma_f^2}} \cdot e^{-\frac{\left(x-\mu_g\right)^2}{2\sigma_g^2}}\right)$$

Multiplying the parts:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\frac{\left(x-\mu_f\right)^2}{2\sigma_f^2}-\frac{\left(x-\mu_g\right)^2}{2\sigma_g^2}}$$

Making the exponent look more familiar gives:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(x-\mu_f\right)^2}{2\sigma_f^2}+\frac{\left(x-\mu_g\right)^2}{2\sigma_g^2}\right)}$$

We can expand the exponent:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{(x-\mu_f)(x-\mu_f)}{2\sigma_f^2}+\frac{(x-\mu_g)(x-\mu_g)}{2\sigma_g^2}\right)}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2-x\mu_f-\mu_f x+\mu_f^2}{2\sigma_f^2}+\frac{x^2-x\mu_g-\mu_g x+\mu_g^2}{2\sigma_g^2}\right)}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2-2x\mu_f+\mu_f^2}{2\sigma_f^2}+\frac{x^2-2x\mu_g+\mu_g^2}{2\sigma_g^2}\right)}$$

Make the exponent have common terms so we can have a consistent denominator:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(x^2-2x\mu_f+\mu_f^2\right)\sigma_g^2}{2\sigma_f^2\sigma_g^2}+\frac{\left(x^2-2x\mu_g+\mu_g^2\right)\sigma_f^2}{2\sigma_g^2\sigma_f^2}\right)}$$

Now we can combine and simplify to a single denominator:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(x^2-2x\mu_f+\mu_f^2\right)\sigma_g^2+\left(x^2-2x\mu_g+\mu_g^2\right)\sigma_f^2}{2\sigma_f^2\sigma_g^2}\right)}$$

Now we expand the terms:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2\sigma_g^2-2x\mu_f\sigma_g^2+\mu_f^2\sigma_g^2+x^2\sigma_f^2-2x\mu_g\sigma_f^2+\mu_g^2\sigma_f^2}{2\sigma_f^2\sigma_g^2}\right)}$$

Group things by factors of x:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2\sigma_g^2+x^2\sigma_f^2-2x\mu_f\sigma_g^2-2x\mu_g\sigma_f^2+\mu_f^2\sigma_g^2+\mu_g^2\sigma_f^2}{2\sigma_f^2\sigma_g^2}\right)}$$

And factor things:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(\sigma_g^2+\sigma_f^2\right)x^2-2\left(\mu_f\sigma_g^2+\mu_g\sigma_f^2\right)x+\mu_f^2\sigma_g^2+\mu_g^2\sigma_f^2}{2\sigma_f^2\sigma_g^2}\right)}$$

Each of the σ's is greater than 0, so we can divide the top and bottom by the coefficient of x^2:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(\frac{\sigma_g^2+\sigma_f^2}{\sigma_g^2+\sigma_f^2}\right)x^2-2\left(\frac{\mu_f\sigma_g^2+\mu_g\sigma_f^2}{\sigma_g^2+\sigma_f^2}\right)x+\frac{\mu_f^2\sigma_g^2+\mu_g^2\sigma_f^2}{\sigma_g^2+\sigma_f^2}}{2\left(\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2+\sigma_f^2}\right)}\right)}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2-2\frac{\mu_f\sigma_g^2+\mu_g\sigma_f^2}{\sigma_g^2+\sigma_f^2}x+\frac{\mu_f^2\sigma_g^2+\mu_g^2\sigma_f^2}{\sigma_g^2+\sigma_f^2}}{2\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2+\sigma_f^2}}\right)}$$

Now we're going to do a little pattern matching trick by looking at the equation of a normal Gaussian:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2-2x\mu+\mu^2}{2\sigma^2}}$$

The math works out that the coefficients of the terms in the exponent will have to be equal. This means that the mean and standard deviation of our product of two Gaussians can be identified as:

$\mu_{fg} = \frac{\mu_f\sigma_g^2+\mu_g\sigma_f^2}{\sigma_g^2+\sigma_f^2}$ and $\sigma_{fg}^2 = \frac{\sigma_f^2\sigma_g^2}{\sigma_g^2+\sigma_f^2} \rightarrow \sigma_{fg} = \sqrt{\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2+\sigma_f^2}}$

Now, we sort of cheated on the pattern matching because we ignored a lot of the constants (e.g. values that don't depend on x). We'll fix that up by subtracting a γ that will represent all the messy constants we ignored. This uses a technique called "completing the square" as described in (11)

$$f(x)g(x) = \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2-2\frac{\mu_f\sigma_g^2+\mu_g\sigma_f^2}{\sigma_g^2+\sigma_f^2}x+\frac{\mu_f^2\sigma_g^2+\mu_g^2\sigma_f^2}{\sigma_g^2+\sigma_f^2}}{2\frac{\sigma_f^2\sigma_g^2}{\sigma_g^2+\sigma_f^2}}\right)}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{x^2-2\mu_{fg}x+\mu_{fg}^2}{2\sigma_{fg}^2}\right)-\gamma}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\left(\frac{\left(x-\mu_{fg}\right)^2}{2\sigma_{fg}^2}\right)-\gamma}$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\frac{\left(x-\mu_{fg}\right)^2}{2\sigma_{fg}^2}-\gamma}$$

Using a basic rule on exponents, we can expand the exponent term to:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{-\frac{\left(x-\mu_{fg}\right)^2}{2\sigma_{fg}^2}} e^{-\gamma}$$

And simplify slightly to:

$$f(x)g(x) = \frac{e^{-\gamma}}{2\pi\sigma_f\sigma_g} e^{-\frac{\left(x-\mu_{fg}\right)^2}{2\sigma_{fg}^2}}$$

Adding and Subtracting Gaussians

How do you combine two functions similar to “adding” and “subtracting?” Many ways are possible, but in many applied areas, the “convolution” operator is used. Terry Tao defines (12) convolution this way:

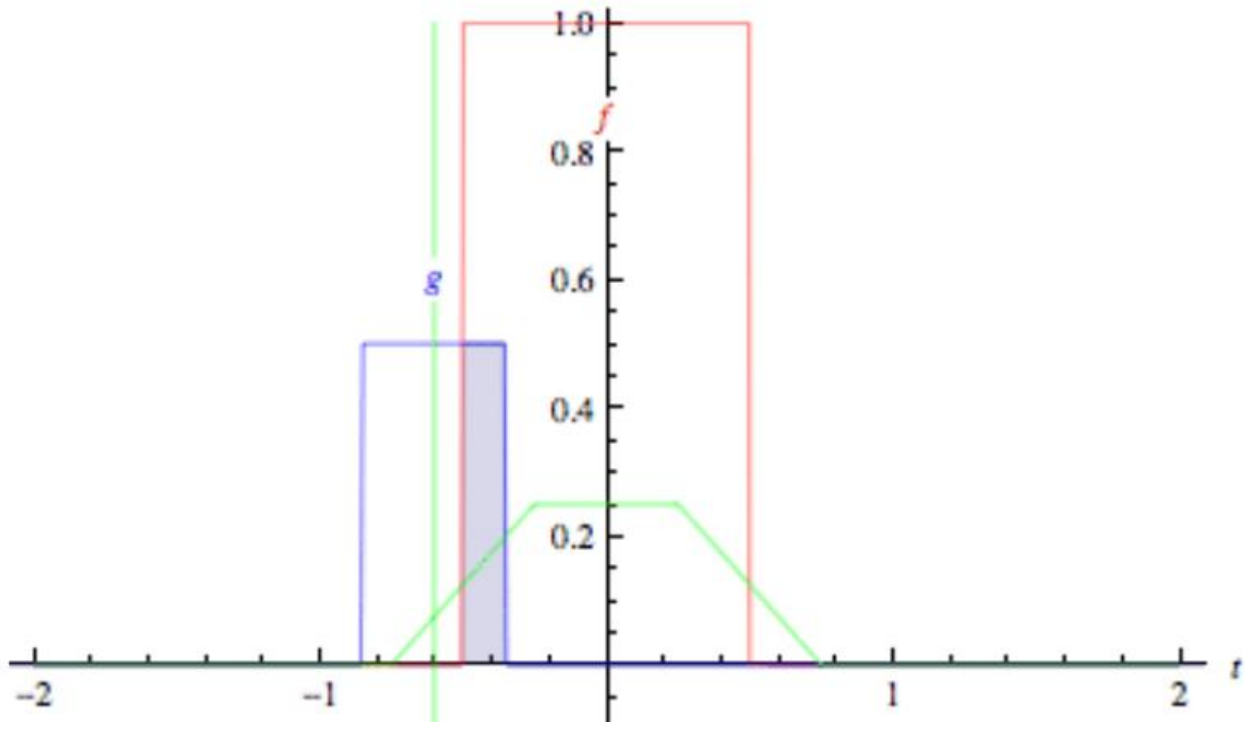
I remember as a graduate student that Ingrid Daubechies frequently referred to convolution by a bump function as "blurring" - its effect on images is similar to what a short-sighted person experiences when taking off his or her glasses (and, indeed, if one works through the geometric optics, convolution is not a bad first approximation for this effect). I found this to be very helpful, not just for understanding convolution per se, but as a lesson that one should try to use physical intuition to model mathematical concepts whenever one can.

More generally, if one thinks of functions as fuzzy versions of points, then convolution is the fuzzy version of addition (or sometimes multiplication, depending on the context). The probabilistic interpretation is one example of this (where the fuzz is a probability distribution), but one can also have signed, complex-valued, or vector-valued fuzz, of course.

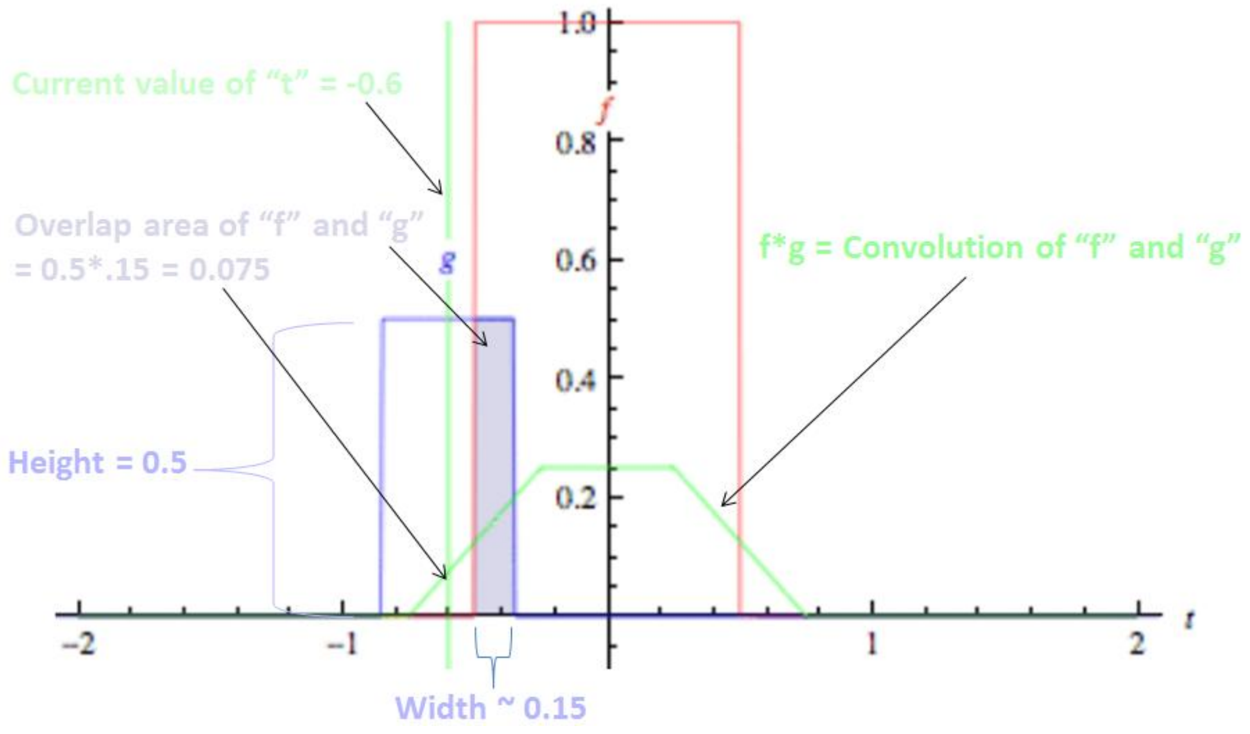
MathWorld (13) defines convolution as:

“[A]n integral that expresses the amount of overlap of one function g as it is shifted over another function f. It therefore ‘blends’ one function with another.”

Fortunately, MathWorld gives pictures to make the concept clearer with two illustrations. The first shows two boxcar functions (e.g. functions that have a fixed value for a specific range):



Above we see that “f” (red) is being convolved with “g” (blue) to produce the green function. In this example, the “g” box is moving from the left to the right. The above graph is a snapshot of this moving process exactly when the middle of “g” was at -0.6. We can measure that the height of “g” is 0.5 and the width of the overlap at this instant in time is approximately 0.15. Therefore, the area that expresses the overlap (gray) is the product of the width multiplied by the height which is 0.5 * 0.15 = 0.075. As you can see, the green curve at t = -0.6 is indeed what we expect (~0.075). The annotated graph looks like this:



A key point about the "convolution" graph is that every single point on it is a measure of the total overlap between the "f" and "g" functions. In other words, it's the total overlapping area from $t = -\infty$ to $t = \infty$.

We'll now use some calculus to express this overlap more formally in order to come up with a formula to compute its value. For a given instant in time x, the convolution of f and g, written as $[f \otimes g](x)$ will express the entire area of overlap of the two functions. Therefore, we already know that the integral will be something like this:

$$[f \otimes g](x) = \int_{-\infty}^{\infty} (\text{overlap of } f \text{ and } g \text{ at } t \text{ when } x \text{ is fixed})\, dt$$

It's easy to get confused by the variables above. Instead of using the normal x to indicate the particular point where we want to compute the convolution, we'll use the Greek letter τ (tau) to indicate that we're evaluating the convolution for a specific point on the t axis. That is, we'll use the similar looking t and τ to denote that we're talking about the same axis:

$$[f \otimes g](t) = \int_{-\infty}^{\infty} (\text{overlap of } f \text{ and } g \text{ at } \tau \text{ when } t \text{ is fixed})\, d\tau$$

It is very important to see that the integral is $d\tau$ and not dt. In other words, t is constant for the entire integral while τ covers the range of the entire line.

Now we must figure out the value of "overlap of f and g at τ when t is fixed" which is what the integral sums. That is, we want to figure out how to calculate the height of the overlapping region for any τ when t is fixed. Once we have the height, we can calculate the infinitesimal area by multiplying it by $d\tau$.

NOTE: The following transition of a convolution definition is a little vague and "hand-wavy." Please feel free to email me if you have a more intuitive definition of how convolution works.

By definition (14), a convolution of two functions "f" and "g" is defined as "the integral of the product of two functions after one is reversed and shifted." If "f" and "g" are Gaussian, we have:

$f(x) = \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{(x-\mu_f)^2}{2\sigma_f^2}}$ and $g(x) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(x-\mu_g)^2}{2\sigma_g^2}}$

Plugging these into the definition of a convolution gives us the following result:

$$[f \otimes g](t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{f\otimes g}} e^{\left(-\frac{(x-\mu_{f\otimes g})^2}{2\sigma_{f\otimes g}^2}\right)}$$

where

$$\mu_{f\otimes g} = \mu_f + \mu_g$$

and

$$\sigma_{f\otimes g} = \sqrt{\sigma_f^2 + \sigma_g^2}$$

which implies

$$\sigma_{f\otimes g}^2 = \left(\sqrt{\sigma_f^2 + \sigma_g^2}\right)^2 = \sigma_f^2 + \sigma_g^2$$

We'll need to prove this. In (10), the author uses Fourier transforms and clever substitutions to come up with the result. Although I've heard of Fourier transforms and their ability to go between frequency and time domains, it was a bit too much to follow. Calin Miron was kind and contacted me with a simpler derivation (15) that I include and expand upon here:

First, we'll head back to the definition of a convolution:

$$[f \otimes g](t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$

Substituting in our Gaussian definitions for "f" and "g":

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{(t-\mu_f-\tau)^2}{2\sigma_f^2}} \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(t-\mu_g)^2}{2\sigma_g^2}} d\tau$$

Collecting constants:

$$= \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2\pi}\sigma_f} \cdot \frac{1}{\sqrt{2\pi}\sigma_g}\right) e^{-\frac{(t-\mu_f-\tau)^2}{2\sigma_f^2}} \cdot e^{-\frac{(\tau-\mu_g)^2}{2\sigma_g^2}} d\tau$$

$$= \int_{-\infty}^{\infty} \left(\frac{1}{2\pi\sigma_f\sigma_g}\right) e^{-\frac{(t-\mu_f-\tau)^2}{2\sigma_f^2}} \cdot e^{-\frac{(\tau-\mu_g)^2}{2\sigma_g^2}} d\tau$$

$$(f \otimes g)(t) = \frac{1}{2\pi\sigma_f\sigma_g} \int_{-\infty}^{\infty} e^{-\frac{(t-\mu_f-\tau)^2}{2\sigma_f^2}} \cdot e^{-\frac{(\tau-\mu_g)^2}{2\sigma_g^2}} d\tau$$

This integral is too complicated, so we'll need to simplify it by making some substitutions. If we look at the numerators, we can see that $(\tau - \mu_g)^2$ is simpler than $(t - \mu_f - \tau)^2$ so let's try making a substitution of:

$$u = \tau - \mu_g$$

We now need to represent $t - \mu_f - \tau$ in terms of u. We'll call this other term y and then solve for it:

$$\begin{aligned}
&y - u = t - \mu_f - \tau \\
&y - (\tau - \mu_g) = t - \mu_f - \tau \\
&y - \tau + \mu_g = t - \mu_f - \tau \\
&y + \mu_g - \tau = t - \mu_f - \tau \\
&y + \mu_g = t - \mu_f \\
&y = t - \mu_f - \mu_g \\
&y = t - (\mu_f + \mu_g)
\end{aligned}$$

As mentioned earlier, we have set

$$\mu_{f\otimes g} = \mu_f + \mu_g$$

This further simplifies y:

$$\begin{aligned}
y &= t - (\mu_f + \mu_g) \\
&= t - \mu_{f\otimes g}
\end{aligned}$$

We now can substitute y and u back into our original integral. However, we have to be careful because the previous integral was $d\tau$ and now the τ's have been substituted. This isn't a problem because both y

and u are just linear shifts of τ which had infinite limits, therefore the limits of integration can remain the same:

$$[f \otimes g](t) = \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\left(t-\mu_f-\tau\right)^2}{2\sigma_f^2}} \cdot e^{-\frac{\left(\tau-\mu_g\right)^2}{2\sigma_g^2}} d\tau$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{(y-u)^2}{2\sigma_f^2}} \cdot e^{-\frac{u^2}{2\sigma_g^2}} du$$

We recall that

$$e^A e^B = e^{A+B}$$

So we can rewrite

$$[f \otimes g](t) = \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{(y-u)^2}{2\sigma_f^2}} \cdot e^{-\frac{u^2}{2\sigma_g^2}} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{(y-u)^2}{2\sigma_f^2}-\frac{u^2}{2\sigma_g^2}} du$$

We can now make a common denominator in the exponent of $2\sigma_f^2\sigma_g^2$ by multiplying by the appropriate numerator on each term:

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2(y-u)^2}{2\sigma_f^2\sigma_g^2}-\frac{\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\left(\frac{\sigma_g^2(y-u)^2}{2\sigma_f^2\sigma_g^2}+\frac{\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

And now we can collect things since we have a common denominator:

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2(y-u)^2+\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}} du$$

Expanding the $(y-u)^2$exponent:

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2(y-u)(y-u)+\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2(y^2-yu-uy+u^2)+\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}}du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2(y^2-2yu+u^2)+\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}}du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_g^2y^2-2\sigma_g^2yu+\sigma_g^2u^2+\sigma_f^2u^2}{2\sigma_f^2\sigma_g^2}}du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_f^2u^2+\sigma_g^2u^2-2\sigma_g^2yu+\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}}du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\left(\sigma_f^2+\sigma_g^2\right)u^2-2\sigma_g^2yu+\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}}du$$

We can now substitute back in $\sigma_{f\otimes g}^2 = \sigma_f^2 + \sigma_g^2$:

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\frac{\sigma_{f\otimes g}^2u^2-2\sigma_g^2yu+\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}}du$$

Breaking up the exponential part:

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{-\left(\frac{\sigma_{f\otimes g}^2u^2-2\sigma_g^2yu}{2\sigma_f^2\sigma_g^2}\right)-\left(\frac{\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}\right)}du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g}\int_{-\infty}^{\infty} e^{\left(-\frac{\sigma_{f\otimes g}^2u^2-2\sigma_g^2yu}{2\sigma_f^2\sigma_g^2}\right)}e^{\left(-\frac{\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}\right)}du$$

Because we are integrating with respect to u, we can treat the $y = t - \left(\mu_f + \mu_g\right)$ as constant:

$$= \frac{1}{2\pi\sigma_f\sigma_g}e^{\left(-\frac{\sigma_g^2y^2}{2\sigma_f^2\sigma_g^2}\right)}\int_{-\infty}^{\infty} e^{\left(-\frac{\sigma_{f\otimes g}^2u^2-2\sigma_g^2yu}{2\sigma_f^2\sigma_g^2}\right)}du$$

Canceling out the σ_g^2 constant on the left of the integral:

$$= \frac{1}{2\pi\sigma_f\sigma_g}e^{\left(-\frac{y^2}{2\sigma_f^2}\right)}\int_{-\infty}^{\infty} e^{\left(-\frac{\sigma_{f\otimes g}^2u^2-2\sigma_g^2yu}{2\sigma_f^2\sigma_g^2}\right)}du$$

We will now use the “completing the square” technique described in (11) for u:

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \left(\sigma_{f\otimes g} u - A\right)^2 + k$$

For some A and k

We can figure out what these values should be by expanding the right side:

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \left(\sigma_{f\otimes g} u - A\right)^2 + k$$

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \left(\sigma_{f\otimes g} u - A\right)\left(\sigma_{f\otimes g} u - A\right) + k$$

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \sigma_{f\otimes g}^2 u^2 - \sigma_{f\otimes g} uA - A\sigma_{f\otimes g} u + A^2 + k$$

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \sigma_{f\otimes g}^2 u^2 - 2\sigma_{f\otimes g} uA + A^2 + k$$

We can simplify by subtracting $\sigma_{f\otimes g}^2 u^2$ from both sides:

$$-2\sigma_g^2 yu = -2\sigma_{f\otimes g} uA + A^2 + k$$

Now, we can eliminate the A^2 part if we set

$$k = -A^2$$

This gives us:

$$-2\sigma_g^2 yu = -2\sigma_{f\otimes g} uA + A^2 - A^2$$

$$-2\sigma_g^2 yu = -2\sigma_{f\otimes g} uA$$

Leading to the simplification:

$$\sigma_g^2 yu = \sigma_{f\otimes g} uA$$

$$\sigma_g^2 y = \sigma_{f\otimes g} A$$

$$\sigma_{f\otimes g} A = \sigma_g^2 y$$

$$A = \frac{\sigma_g^2 y}{\sigma_{f\otimes g}}$$

Thus, we can go back to our original equation:

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \left(\sigma_{f\otimes g} u - A\right)^2 + k$$

And substitute back in our values:

$$\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu = \left(\sigma_{f\otimes g} u - A\right)^2 - A^2$$

$$= \left(\sigma_{f\otimes g} u - \frac{\sigma_g^2 y}{\sigma_{f\otimes g}}\right)^2 - \left(\frac{\sigma_g^2 y}{\sigma_{f\otimes g}}\right)^2$$

We can now pick up where we left off:

$$(f \otimes g)(t) = \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} \int_{-\infty}^{\infty} e^{\left(-\frac{\sigma_{f\otimes g}^2 u^2 - 2\sigma_g^2 yu}{2\sigma_f^2\sigma_g^2}\right)} du$$

And substitute in our completed square

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} \int_{-\infty}^{\infty} e^{\left(-\frac{\left(\sigma_{f\otimes g} u - A\right)^2 - A^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

Breaking apart the exponent:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g} u - A\right)^2 + A^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g} u - A\right)^2}{2\sigma_f^2\sigma_g^2} + \frac{A^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g} u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} e^{\left(\frac{A^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We recall that

$$A = \frac{\sigma_g^2 y}{\sigma_{f\otimes g}}$$

And thus does not depend on u and thus the rightmost part of the integral, $e^{\left(\frac{A^2}{2\sigma_f^2\sigma_g^2}\right)}$, can be moved outside the integral:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} e^{\left(\frac{A^2}{2\sigma_f^2\sigma_g^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g} u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We can start to substitute the value of A:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} e^{\left(\frac{\left(\frac{\sigma_g^2 y}{\sigma_{f\otimes g}}\right)^2}{2\sigma_f^2\sigma_g^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} e^{\left(\frac{\left(\frac{\sigma_g^4 y^2}{\sigma_{f\otimes g}^2}\right)}{2\sigma_f^2\sigma_g^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} e^{\left(\frac{\sigma_g^4 y^2}{\sigma_{f\otimes g}^2 2\sigma_f^2\sigma_g^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

And simplify:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}\right)} e^{\left(\frac{\sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

Grouping together the exponential constants:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2}{2\sigma_f^2}+\frac{\sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

Creating a common denominator for the exponential constant:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(-\frac{y^2\sigma_{f\otimes g}^2}{2\sigma_f^2\sigma_{f\otimes g}^2}+\frac{\sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-y^2\sigma_{f\otimes g}^2+\sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We recall that:

$$\sigma_{f\otimes g}^2 = \left(\sqrt{\sigma_f^2+\sigma_g^2}\right)^2 = \sigma_f^2+\sigma_g^2$$

And substitute this back in to the constant part:

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-y^2\left(\sigma_f^2+\sigma_g^2\right)+\sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-(\sigma_{f\otimes g}u-A)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-y^2\sigma_f^2 - y^2\sigma_g^2 + \sigma_g^2 y^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-y^2\sigma_f^2}{2\sigma_f^2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-y^2}{2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We can eliminate y by recalling that $y = t - \mu_{f\otimes g}$, so we obtain:

$$[f \otimes g](t) = \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We're now getting close to what we expected. We remember that a normalized Gaussian with mean μ_f and standard deviation σ_f is defined as:

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{\left(t-\mu_f\right)^2}{2\sigma_f^2}}$$

This very closely matches the constant multiplier of our integral. We're just off by a constant factor. We need to figure out this factor to make the convolution properly normalized. This will require us to calculate:

$$I = \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

$$= \int_{-\infty}^{\infty} e^{\left(-\frac{\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$

We once again simplify the numerator by a substitution:

$$w = \sigma_{f\otimes g}u - A$$

Again, we recall that we were du so we'll need to now change to dw. We can take the derivative of each side:

$$dw = \sigma_{f\otimes g}\, du$$

This substitution won't change the limits of integration because they were infinite before and all we did was a linear shift. This gives us:

$$I = \int_{-\infty}^{\infty} e^{\left(-\frac{\left(\sigma_{f\otimes g}u - A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du$$
$$= \int_{-\infty}^{\infty} e^{\left(-\frac{w^2}{2\sigma_f^2\sigma_g^2}\right)} \frac{dw}{\sigma_{f\otimes g}}$$

Factoring out a constant:

$$= \frac{1}{\sigma_{f\otimes g}} \int_{-\infty}^{\infty} e^{\left(-\frac{w^2}{2\sigma_f^2\sigma_g^2}\right)} dw$$

We'd like to further simplify the integral by making yet another substitution. We notice all the squared terms and propose:

$$z = \sqrt{\frac{w^2}{2\sigma_f^2\sigma_g^2}}$$
$$= \frac{w}{\sqrt{2}\sigma_f\sigma_g}$$

Our existing integral is currently with respect to dw. We'll need to make it with respect to dz. Again, this won't affect the limits of integration, but it will affect the derivative as:

$$dz = \frac{dw}{\sqrt{2}\sigma_f\sigma_g}$$

Substituting this back into the integral gives us:

$$I = \frac{1}{\sigma_{f\otimes g}} \int_{-\infty}^{\infty} e^{\left(-\frac{w^2}{2\sigma_f^2\sigma_g^2}\right)} dw$$
$$= \frac{1}{\sigma_{f\otimes g}} \int_{-\infty}^{\infty} e^{(-z^2)} dz \left(\sqrt{2}\sigma_f\sigma_g\right)$$

Rearranging the constant:

$$= \frac{\sqrt{2}\sigma_f\sigma_g}{\sigma_{f\otimes g}} \int_{-\infty}^{\infty} e^{(-z^2)} dz$$

Now, we can use the result that we derived on page 38, namely that

$$\int_{-\infty}^{\infty} e^{-z^2} dz = \sqrt{\pi}$$

To give us:

$$
\begin{aligned}
I &= \frac{\sqrt{2}\sigma_f\sigma_g}{\sigma_{f\otimes g}}\int_{-\infty}^{\infty} e^{(-z^2)}dz \\
&= \frac{\sqrt{2}\sigma_f\sigma_g}{\sigma_{f\otimes g}}\left(\sqrt{\pi}\right) \\
&= \frac{\sqrt{2\pi}\sigma_f\sigma_g}{\sigma_{f\otimes g}}
\end{aligned}
$$

We can now go back to our previous result and substitute in I:

$$
\begin{aligned}
[f\otimes g](t) = {} & \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \int_{-\infty}^{\infty} e^{\left(\frac{-\left(\sigma_{f\otimes g}u-A\right)^2}{2\sigma_f^2\sigma_g^2}\right)} du \\
= {} & \frac{1}{2\pi\sigma_f\sigma_g} e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \left(\frac{\sqrt{2\pi}\sigma_f\sigma_g}{\sigma_{f\otimes g}}\right) \\
= {} & \left(\frac{\sqrt{2\pi}\sigma_f\sigma_g}{\sigma_{f\otimes g}}\right)\left(\frac{1}{2\pi\sigma_f\sigma_g}\right) e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \\
= {} & \left(\frac{\sqrt{2\pi}\sigma_f\sigma_g}{2\pi\sigma_f\sigma_g\sigma_{f\otimes g}}\right) e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \\
= {} & \left(\frac{\sqrt{2\pi}}{2\pi\sigma_{f\otimes g}}\right) e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \\
= {} & \left(\frac{\sqrt{2\pi}}{\sqrt{2\pi}\sqrt{2\pi}\sigma_{f\otimes g}}\right) e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)} \\
= {} & \left(\frac{1}{\sqrt{2\pi}\sigma_{f\otimes g}}\right) e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)}
\end{aligned}
$$

Thus, we *finally* have proved the formula for the convolution of two Gaussian functions:

$$
[f\otimes g](t) = \frac{1}{\sqrt{2\pi}\sigma_{f\otimes g}} e^{\left(\frac{-\left(t-\mu_{f\otimes g}\right)^2}{2\sigma_{f\otimes g}^2}\right)}
$$

Works Cited

1. **Herbrich, Ralf and Graepel, Thore.** TrueSkill: A Bayesian Skill Rating System. [Online] June 2006. http://research.microsoft.com/apps/pubs/default.aspx?id=67956.

2. **Herbrich, Ralf.** Gamefest 2007 : LIVE : Ranking and Matchmaking TrueSkill Revealed. [Online] 2007. http://www.microsoft.com/downloads/en/confirmation.aspx?familyId=1acc9bf7-920d-477b-a7b1-4945b3cb04dd&displayLang=en.

3. **Alpaydin, Ethem.** *Introduction to Machine Learning (Second Edition).* Cambridge, Massachusetts : The MIT Press, 2010.

4. **Azad, Kalid.** An Intuitive (and Short) Explanation of Bayes' Theorem. *BetterExplained.* [Online] May 6, 2007. http://betterexplained.com/articles/an-intuitive-and-short-explanation-of-bayes-theorem/.

5. *Factor Graphs and the Sum-Product Algorithm.* **Kschischang, Frank R., Frey, Brendan J. and Loeliger, Hans-Andrea.** 2, s.l. : IEEE Transactions on Information Theory, 2001, Vol. 47.

6. **Weisstein, Eric W.** Normal Sum Distribution. *MathWorld--A Wolfram Web Resource.* [Online] http://mathworld.wolfram.com/NormalSumDistribution.html.

7. **Herbrich, Ralf.** On Gaussian Expectation Propagation. [Online] July 2005. http://research.microsoft.com/apps/pubs/default.aspx?id=74554.

8. **Herbrich, Ralf and Graepel, Thore.** TrueSkill™ Ranking System: Details. *Microsoft Research.* [Online] http://research.microsoft.com/en-us/projects/trueskill/details.aspx.

9. **Eracleous, Mike.** Moments of the Gaussian Distribution and Associated Integrals. [Online] 2004. http://www.astro.psu.edu/~mce/A451_2/A451/downloads/notes0.pdf.

10. **Bromiley, P A.** Products and Convolutions of Gaussian Distributions. [Online] November 27, 2003. http://www.tina-vision.net/tina-knoppix/tina-memo/2003-003.pdf.

11. Completing the Square. [Online] Wikipedia, 2010. [Cited: November 30, 2010.] http://en.wikipedia.org/wiki/Completing_the_square.

12. **Tao, Terry.** What is Convolution Intuitively? *MathOverflow.* [Online] November 18, 2009. [Cited: December 7, 2010.] http://mathoverflow.net/questions/5892/what-is-convolution-intuitively/5916#5916.

13. **Weisstein, Eric W.** Convolution. *MathWorld.* [Online] A Wolfram Web Resource, August 18, 2008. [Cited: December 7, 2010.] http://mathworld.wolfram.com/Convolution.html.

14. Convolution. *Wikipedia.* [Online] November 18, 2010. [Cited: November 30, 2010.] http://en.wikipedia.org/wiki/Convolution.

15. **Miron, Calin.** *Convolution of two Gaussians.* November 30, 2010.

16. **Bishop, Christopher.** Embracing Uncertainty: The new machine intellgience. [Online] February 25, 2010. http://scpro.streamuk.com/uk/player/Default.aspx?wid=7739.