

Práctica 4. Flujo de inferencia.

Docente: Gustavo Landfried

Inferencia Bayesiana Causal 1
1er cuatrimestre 2025
ECyT - UNSAM

Índice

1. Algoritmo suma-producto: inferencia por pasaje de mensajes.	3
1.1. Verificar en qué condiciones se abre y se cierra el flujo de inferencia en las estructuras causales elementales	4
2. La clase Gaussiana	4
2.1. Propiedades de la distribución Gaussiana	5
2.2. Implementar la clase Gaussiana	5
3. Estimación de habilidad por pasaje de mensajes	6
3.1. Mensajes descendentes	7
3.2. Menajes ascendentes	9

Introducción

Cuando tenemos 5 variables, existen $5! = 120$ posibles especificaciones alternativas de distribución de probabilidad conjunta.

$$P(l, e, t, r, a) = P(l)P(e|l)P(t|l, e)P(r, a|l, e, t) = \cancel{P(t)} \frac{\cancel{P(t, e)}}{P(t)} \frac{\cancel{P(t, e, t)}}{\cancel{P(t, e)}} \frac{P(l, e, t, r, a)}{\cancel{P(t, e, t)}}$$

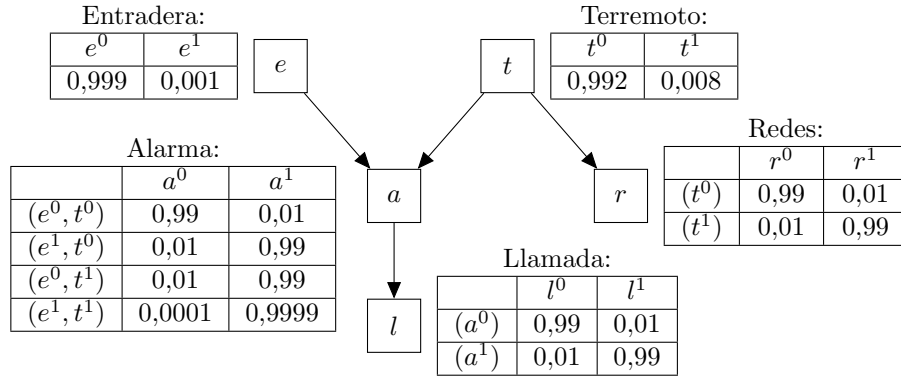
¿Cuál de todas las posibles especificaciones utilizar? Supongamos que la siguiente es la interpretación de las variables.

La **alarma** (a) de una casa se activa cuando alguien **entra** (e) sin apagarla. Si la alarma se activa, la dueña recibe una **llamada** (l) de la central de seguridad. La alarma se puede activar por otros motivos, como los **terremotos** (t) que no son infrecuentes en la ciudad. Siempre que se produce un terremoto, en las **redes** (r) se habla casi exclusivamente de eso. En la ciudad ocurren 3 terremotos leves por año y hay intentos por entrar a la casa 1 vez cada 3 años.

¿Cómo podemos definir la distribución de probabilidad condicional $P(t|l, e)$? No parece nada fácil. Pero si especificamos la distribución conjunta siguiendo las relaciones causales probabilísticas entre las variables,

$$P(l, e, t, r, a) = P(a|e, t)P(e)P(l|a)P(t)P(r|t)$$

cada una de las distribuciones condicionales empleadas tiene una interpretación intuitiva relativamente fácil de determinar. Las siguientes distribuciones de probabilidad condicional siguen la descripción causal del problema. Notar que si bien la descripción de algunos eventos es deterministas, tal como “siempre que la alarma se activa la dueña recibe una llamada”, al especificar la distribución de probabilidad condicional no concentrada toda la creencia en la opción indicada, $P(l = 1|a = 1) = 1$, para admitir cierto ruido observacional.



Ahora sí, podemos calcular cualquier posterior del modelo como el cociente de dos marginales. Para determinar la probabilidad del terremoto dado que hubo una llamada y sabemos que alguien entró, se puede calcular como

$$P(t|l, e) = \frac{P(t, l, e)}{P(l, e)} = \frac{\sum_{a, r} P(t, r, e, a, l)}{\sum_{t, a, r} P(t, r, e, a, l)} = \frac{\sum_{a, r} P(t)P(r|t)P(e)P(a|e, t)P(l|a)}{\sum_{t, a, r} P(t)P(r|t)P(e)P(a|e, t)P(l|a)}$$

Es decir, todas nuestras preguntas (condicionales) se reducen a calcular marginales de la conjunta. Calcular marginales es “predecir con la contribución de todas las hipótesis”, lo cual es computacionalmente costoso. Por ejemplo, para predecir la activación de la alarma, necesitamos integrar sobre el resto de las variables del modelo,

$$P(a) = \sum_t \sum_r \sum_e \sum_l P(t, r, e, a, l) = \sum_t \sum_r \sum_e \sum_l P(t)P(r|t)P(e)P(a|e, t)P(l|a)$$

Si lo pensamos en términos computacionales, calcular la marginal $P(a)$ suponer realizar 4 ciclos anidados, uno adentro del otro.

```
suma = 0
for t in Hipotesis.T:
  for r in Hipotesis.R:
    for e in Hipotesis.E:
      for l in Hipotesis.L:
        suma ...
```

En este caso son solo 4 variables y cada una tiene 2 posibles valores, por lo que la cantidad de iteraciones no resulta ser tan alta, solo $2^4 = 16$. Sin embargo, como las iteraciones producidas por ciclos anidados crecen de forma exponencial, cuando el modelo tiene unas pocas variables más con varios posibles valores, su cálculo puede volverse prohibitivo.

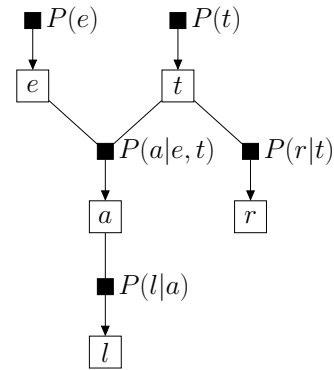
La estructura del modelo causal, sin embargo, permite desanidar ciclos, y eliminar otros. Por ejemplo, en el caso de la marginal de $P(a)$,

$$\begin{aligned} P(a^*) &= \sum_e \sum_t \sum_r \sum_l P(t)P(r|t)P(e)P(a^*|e, t)P(l|a^*) \\ &= \left(\sum_l P(l|a^*) \right) \left(\sum_e \sum_t \sum_r P(e)P(t)P(a^*|t, e)P(r|t) \right) \\ &= \underbrace{\left(\sum_l P(l|a^*) \right)}_1 \left(\sum_e \sum_t P(e)P(t)P(a^*|t, e) \underbrace{\sum_r P(r|t)}_1 \right) = \sum_e \sum_t P(e)P(t)P(a^*|t, e) \end{aligned}$$

podemos reducir el cómputo a solo dos ciclos anidados.

1. Algoritmo suma-producto: inferencia por pasaje de mensajes.

La aplicación de las (dos) reglas de la probabilidad pueden descomponerse como pasaje de mensajes entre los nodos de la red causal que especifica su probabilidad conjunta.



(a) Factor graph. Nodos: variables y funciones. Ejes: la variable es parámetro de la función.

$$\begin{aligned} P(x) &= \prod_{f \in v(x)} m_{f \rightarrow x}(x) \\ &\text{La marginal de } x \text{ es producto de todos los mensajes entrantes} \\ m_{x \rightarrow f}(x) &= \prod_{h \in v(x) \setminus \{f\}} m_{h \rightarrow x}(x) \\ &\text{Las variables envían para adelante el producto de los mensajes que recibe de atrás} \\ m_{f \rightarrow x}(x) &= \sum_{\mathbf{h}} \left(f(\mathbf{h}, x) \prod_{h \in v(f) \setminus \{x\}} m_{h \rightarrow f}(h) \right) \end{aligned}$$

Los factores envían para adelante el producto de lo que reciben de atrás con signo mismo, integrando por las variables de atrás

Figura 1: Sum-product algorithm, pasaje de mensajes en factor graphs.

Los *factor graph* (figura 1a) son un tipo de especificación gráfica alternativa a la redes bayesianas, en las cuales la distribución de probabilidad condicional pasan a formar parte como nodos del

grafo, vinculadas a las variables que son sus parámetros. El uso de esta especificación gráfica nueva está justificado en el hecho de que las reglas de la probabilidad pueden computarse localmente como pasaje de mensajes entre los nodos de la red.

1.1. Verificar en qué condiciones se abre y se cierra el flujo de inferencia en las estructuras causales elementales

La estructura causal determina el flujo de inferencia entre las variables de la red causal. Decimos que un flujo de inferencia entre una variable x y una variable y se encuentra abierto si observar una de ellas nos permite actualizar la creencia respecto de la otra que permanece no observada. Y decimos que un flujo de inferencia entre una variable x y una variable y se encuentra cerrado si la observación de una de ellas no modifica la creencia respecto de la otra variable que permanece no observada.

Existen 3 tipos de estructuras causales elementales. Las estructuras causales elementales se definen entre 3 variables, y la clase de estructura causal se define por el tipo de variable intermedia m . Llamamos *pipe* a las estructuras causales elementales en las que la variable intermedia preserva la dirección causal ($x \rightarrow m \rightarrow y$), o viceversa. Llamamos *fork* o *co-founder* a las estructuras causales elementales en las que la variable intermedia es causa común de las dos variables de interés ($x \leftarrow m \rightarrow y$). Y llamamos *collider* a las estructuras causales elementales en las que la variable intermedia está causada simultáneamente por ambas variables de interés, ($x \rightarrow m \leftarrow y$)

La estructura causal de este ejercicio contiene los 3 tipos de estructuras causales elementales. En este ejercicio queremos determinar si observar la variable intermedia abre o cierra el flujo de inferencia. En resumen, queremos determinar si las siguientes igualdades valen siempre o si la igualdad no necesariamente se preserva.

		Intermedio no observable	Intermedio observable
<i>Pipe</i> {	$e \rightarrow a \rightarrow l$	$P(l) \stackrel{?}{=} P(l e)$	$P(l a) \stackrel{?}{=} P(l e, a)$
	$l \leftarrow a \leftarrow t$	$P(t) \stackrel{?}{=} P(t l)$	$P(t a) \stackrel{?}{=} P(t e, l)$
<i>Fork / Co-founder</i> {	$a \leftarrow t \rightarrow r$	$P(r) \stackrel{?}{=} P(r a)$	$P(r t) \stackrel{?}{=} P(r e, a)$
	$e \rightarrow a \leftarrow t$	$P(t) \stackrel{?}{=} P(t e)$	$P(t a) \stackrel{?}{=} P(t e, a)$
<i>Collider</i> {	\downarrow		
	l		$P(t l) \stackrel{?}{=} P(t e, l)$

Por ejemplo, ¿Saber que alguien entró a la casa sin apagar la alarma modifica nuestra creencia sobre la llamada? Podemos fácilmente intuir que sí, que $P(l) \neq P(l|e)$, pues la probabilidad de que llamen $P(l)$ suele ser baja, pero si alguien entra sin apagar la alarma, es muy probable que llamen $P(l|e)$. En este caso, no estamos observando (o condicionado) por la variable del medio y vemos que el flujo de información permanece abierto.

En cambio, ¿qué pasa si conocemos el valor de la variable del medio?. Dado que ya sabemos que la alarma se activó, ¿saber que alguien entró a la casa sin apagar la alarma modifica nuestra creencia sobre la llamada? En este caso podemos intuir que no, que $P(l|a) = P(l|a, e)$, pues conocer el motivo por el cual la alarma se prendió no modifica la probabilidad de llamada, que depende exclusivamente de la alarma $P(l|a) = P(l|a, \ell) = P(l|a, f)$.

Responder cada una de las preguntas enumeradas en la tabla.

2. La clase Gaussiana

En esta sección revisamos primero la propiedades para realizar inferencia exacta mediante pasaje de mensajes en el modelo de habilidad y luego implementamos una clase de python **Gaussian**

con los métodos $+$, $-$, $*$, $/$, $>$, que serán usados en el siguiente ejercicio como motor de inferencia.

2.1. Propiedades de la distribución Gaussiana

En esta sección enumeramos las propiedades que necesitamos para realizar inferencia exacta usando el modelo derivar los mensajes exactos y aproximados que surgen del *sum-product algorithm*. La primera propiedad establece que el producto de dos distribuciones gaussianas, ambas evaluadas en el mismo punto x , pueden expresarse como la producto de otras dos distribuciones gaussianas con sólo una de ellas evaluada en x .

$$\mathcal{N}(x|\mu_1, \sigma_1^2)\mathcal{N}(x|\mu_2, \sigma_2^2) \stackrel{??}{=} \mathcal{N}(\mu_1|\mu_2, \sigma_1^2 + \sigma_2^2)\mathcal{N}(x|\mu_*, \sigma_*^2) \quad (\text{Producto de gaussianas})$$

con $\sigma_*^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}$ y $\mu_* = \left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right)\sigma_*^2$. Algo similar ocurre con la división de dos distribuciones gaussianas, ambas evaluadas en el mismo punto x .

$$\mathcal{N}(x|\mu_1, \sigma_1^2)/\mathcal{N}(x|\mu_2, \sigma_2^2) \stackrel{??}{\propto} \mathcal{N}(x|\mu_{\div}, \sigma_{\div}^2)/\mathcal{N}(\mu_1|\mu_2, \sigma_1^2 + \sigma_2^2) \quad (\text{División de gaussianas})$$

con $\sigma_{\div}^2 = \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)^{-1}$ y $\mu_{\div} = \left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)\sigma_{\div}^2$. La función indicadora $\mathbb{I}(\cdot = \cdot)$ vale 1 cuando la igualdad es verdadera y 0 en caso contrario. Se usa para representar distribuciones de probabilidad de variables discretas no aleatorias, como el resultado de las partidas dada la diferencia de desempeños $p(r|d)$. De la misma forma, la función delta de Dirac $\delta(\cdot = \cdot)$ se usa para representar distribuciones de probabilidad de variables continuas no aleatoria, es la diferencia de desempeños dados los rendimientos de los agentes $p(d|\mathbf{p})$. Cuando permite remplazar variable dentro de una integral,

$$\iint \delta(x = h(y, z))f(x)g(y) dx dy = \int f(h(y, z))g(y)dy \quad (\text{Función delta de dirac})$$

la dimensionalidad del problema se reduce. Usaremos además las propiedades que se derivan de la simetría de gaussianas.

$$\mathcal{N}(x|\mu, \sigma^2) = \mathcal{N}(\mu|x, \sigma^2) = \mathcal{N}(-\mu|-x, \sigma^2) = \mathcal{N}(-x|-\mu, \sigma^2) \quad (\text{Simetría de gaussianas})$$

La estandarización de la gaussiana,

$$\mathcal{N}(x|\mu, \sigma^2) = \mathcal{N}((x - \mu)/\sigma|0, 1) \quad (\text{Estandarización de gaussianas})$$

La igualdad entre la distribución gaussiana y la derivada de la su acumulada,

$$\frac{\partial}{\partial x}\Phi(x|\mu, \sigma^2) = \mathcal{N}(x|\mu, \sigma^2) \quad (\text{Derivada de la gaussiana acumulada})$$

que vale por definición. La simetría de la distribución gaussiana acumulada.

$$\Phi(0|\mu, \sigma^2) = 1 - \Phi(0|-\mu, \sigma^2) \quad (\text{Simetría de la gaussiana acumulada})$$

2.2. Implementar la clase Gaussian

Implementar la clase **Gaussian** para realizar inferencia por pasaje de mensajes en el modelo de estimación de habilidad. Se representa mediante dos parámetros, la media y el desvío estándar.

```
N1 = Gaussian(mu = 1.0, sigma = 1.0); N2 = Gaussian(1.0, 2.0)
```

Code 1: Inicialización de distribuciones gaussianas

La clase sobre escribe los operadores suma (+), resta (-), producto (*) y división (/) con las principales propiedades requeridas para computar las distribuciones marginales en el modelo TrueSkill Through Time.

$$\mathcal{N}(x|\mu_1, \sigma_1^2)\mathcal{N}(x|\mu_2, \sigma_2^2) \stackrel{??}{\propto} \mathcal{N}(x|\mu_*, \sigma_*^2) \quad (\text{N1} * \text{N2})$$

$$\mathcal{N}(x|\mu_1, \sigma_1^2)/\mathcal{N}(x|\mu_2, \sigma_2^2) \stackrel{??}{\propto} \mathcal{N}(x|\mu_{\div}, \sigma_{\div}^2) \quad (\text{N1} / \text{N2})$$

$$\iint \delta(t = x + y)\mathcal{N}(x|\mu_1, \sigma_1^2)\mathcal{N}(y|\mu_2, \sigma_2^2)dxdy \stackrel{??}{=} \mathcal{N}(t|\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) \quad (\text{N1} + \text{N2})$$

$$\iint \delta(t = x - y)\mathcal{N}(x|\mu_1, \sigma_1^2)\mathcal{N}(y|\mu_2, \sigma_2^2)dxdy \stackrel{??}{=} \mathcal{N}(t|\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2) \quad (\text{N1} - \text{N2})$$

Aunque estas propiedades son ampliamente conocidas, adjuntamos sus demostraciones completas en el anexo.

Implementar los métodos de la clase `Gaussian` que se encuentra disponible en el archivo `Gaussian.py`

3. Estimación de habilidad por pasaje de mensajes

Conocer cómo aprenden las personas a lo largo del tiempo es importante en muchos aspectos de la vida, especialmente en el sistema educativo, deportivo y laboral. Dado que la habilidad es una variable oculta, debemos estimarla a partir de sus consecuencias observables directas: el éxito o fracaso en la resolución de problemas y competencias. Considerar sólo la frecuencia de éxitos como indicador de la habilidad de los individuos puede conducir a aproximaciones erróneas, fundamentalmente porque el valor de un éxito o un fracaso depende de la dificultad de los desafíos. Por esta razón, todos los modelos de habilidad ampliamente utilizados consideran que el resultado de la resolución de problemas se produce en relación a otra entidad oculta, que puede ser la habilidad de la persona contra la que compiten o la dificultad intrínseca del problema. Arpad Elo [?] desarrolla en la década del 60' una metodología de estimación de habilidad para la Federación de Ajedrez de los Estados Unidos (USCF), adoptada hasta el día de hoy por la Federación Internacional de Ajedrez (FIDE) ¹.

Aplicado a ajedrez, el modelo causal Elo supone que las habilidades (*skills*) ocultas de dos agentes i y j , $s_i, s_j \in \mathbb{R}$, generan el resultado observable $r \in \{\text{Ganar}, \text{Perder}\}$ a través de la diferencia $d \in \mathbb{R}$ de sus desempeños aleatorios $p_i, p_j \in \mathbb{R}$. En la figura 2 especificamos la versión “gaussiana” del modelo Elo (descrita en el trabajo de Herbrich et. al [?]) utilizando la notación de redes bayesianas.

¹Los primeros modelos causales, propuestos hace casi un siglo por Thurstone y Zermelo, ya suponían que la probabilidad de un resultado observado r depende del rendimiento p (*performance*) del agente i y de su oponente j , $P(r|p_i, p_j)$. El campo siguió progresando con los trabajos de Bradley y Terry y Mosteller, que permitió a Arpad Elo [?] desarrollar su metodología.

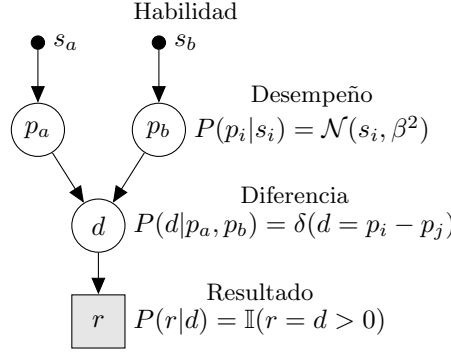


Figura 2: Modelo generativo en el que las habilidades causan los resultados observables a través de la diferencia de los desempeños ocultos, $d = p_i - p_j$, ambas variables aleatorias centradas en la verdadera habilidad, $p \sim \mathcal{N}(s, \beta^2)$. Quien haya obtenido mayor rendimiento gana, $r = (d > 0)$. Las variables observables se pintan de gris, las ocultas en blanco, y las constantes se muestran como puntos negros.

El modelo causal supone que las personas exhiben desempeños aleatorios, que varían alrededor de su verdadera habilidad con un desvío estándar β , $\mathcal{N}(p | s, \beta^2)$. Una vez que se generan los desempeños, el resultado queda determinado pues, según el modelo, gana la persona que obtuvo mayor desempeño, $r = (p_i > p_j)$. El parámetro β^2 , es el mismo para todos los agentes, y actúa como la escala de las estimaciones: habilidades a un β de distancia, $s_i - s_j = \beta$, implica 76 % de probabilidad de ganar, $P(p_i > p_j | s_i, s_j) \approx 0,76$, independientemente del valor absoluto de las habilidades s_i y s_j .

En presencia de equipos, el modelo supone que el desempeño de los equipos t es la suma de los desempeños de sus integrantes, y que el equipo con mayor desempeño gana, $r = (t_i > t_j)$. En la figura 3 mostramos la factorización gráfica del modelo Elo que incorpora equipos. En este

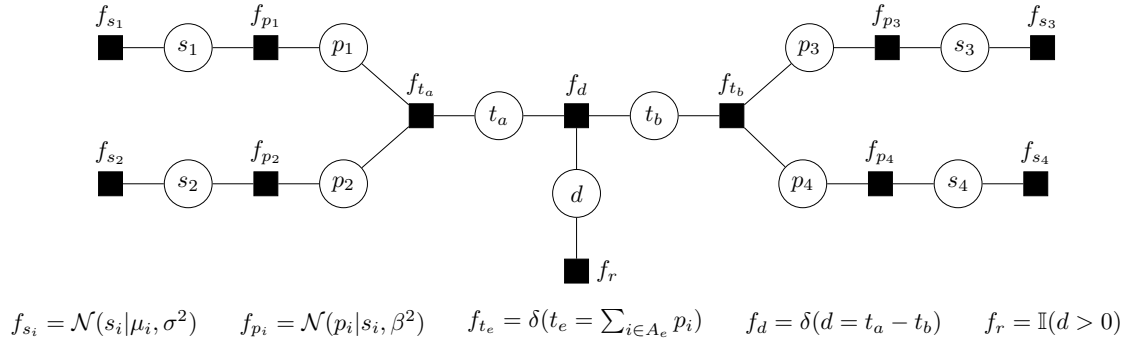


Figura 3: Factorización gráfica de una partida con dos equipos de dos jugadores. El modelo Elo incorpora una variable nueva, t , que modela el desempeño de los equipos.

ejemplo, tenemos dos equipos con dos jugadores cada uno. Toda partida con dos equipos tiene solución analítica exacta.

3.1. Mensajes descendentes

En esta sección mostramos los pasos para calcular la evidencia exacta (mensajes descendentes) de una partida con dos equipos. Sólo necesitamos el *sum-product algorithm* y las propiedades arriba mencionadas. Empezaremos primero con los mensajes “descendentes”, desde los priors al resultado, hasta calcular la evidencia, y seguiremos con los mensajes “ascendentes”, desde el resultado observado a los priors, hasta calcular el posterior de cada agente. Siguiendo el ?? del *sum-product algorithm* y la factorización del modelo presentado en la figura 3, podemos ver que

los mensajes que envían los factores de habilidad f_{s_i} a la variable s_i no son otra cosa más que el prior.

$$m_{f_{s_i} \rightarrow s_i}(s_i) = \mathcal{N}(s_i | \mu_i, \sigma_i^2) \quad (\mathbf{s1})$$

```
s1 = Gaussian(); s2 = Gaussian(); s3 = Gaussian(); s4 = Gaussian()
```

Code 2: Prior de la habilidades

Tenemos acceso a este mensaje llamando al atributo `prior` de la clase `Player`. Podemos ver también, siguiendo el ?? del *sum-product algorithm* y la factorización del modelo, que el mensaje que envía la variable s_i al factor rendimiento f_{p_i} es nuevamente el prior. Debido a que es trivial calcular los mensajes que envían las variables (siempre es el producto de los mensajes que reciben de atrás), vamos a evitar escribirlos. Veamos entonces el mensaje que envían los factores rendimiento f_{p_i} a su variable p_i .

$$\begin{aligned} m_{f_{p_i} \rightarrow p_i}(p_i) &= \int \mathcal{N}(p_i | s_i, \beta^2) \mathcal{N}(s_i | \mu_i, \sigma_i^2) ds_i \\ &= \iint \delta(p_i = s_i + o) \underbrace{\mathcal{N}(o | 0, \beta^2)}_{\text{ruido}} \mathcal{N}(s_i | \mu_i, \sigma_i^2) ds_i do \quad (\mathbf{p1} = \mathbf{s1} + \mathbf{o}) \\ &= \mathcal{N}(p_i | \mu_i, \beta^2 + \sigma_i^2) \end{aligned}$$

Hemos introducido una variable extra al modelo, para poder representar al desempeño p_i de un jugador como la suma de su habilidad oculta s_i y un ruido aleatorio centrado en 0, o . De esta forma podemos ver que el operador suma (+) que implementamos en nuestra clase `Gaussian`, aplicado entre el prior de s_i y la variable aleatoria o codifica el mensaje que el factor f_{p_i} envía a la variable p_i .

```
o = Gaussian(0, 1) # ruido
p1 = s1 + o; p2 = s2 + o; p3 = s3 + o; p4 = s4 + o
```

Code 3: Computando el desempeño individual a priori

El mensaje que envían los factores equipos f_{t_e} a la variable equipo t_e es una integral sobre todas las variables de desempeño individual,

$$\begin{aligned} m_{f_{t_e} \rightarrow t_e}(t_e) &= \iint \delta(t_e = p_i + p_j) \mathcal{N}(p_i | \mu_i, \beta^2 + \sigma_i^2) \mathcal{N}(p_j | \mu_j, \beta^2 + \sigma_j^2) dp_i dp_j \\ &= \mathcal{N}(t_e | \underbrace{\mu_i + \mu_j}_{\mu_e}, \underbrace{2\beta^2 + \sigma_i^2 + \sigma_j^2}_{\sigma_e^2}) \quad (\mathbf{ta} = \mathbf{p1} + \mathbf{p2}) \end{aligned}$$

donde la función delta de Dirac impone la restricción de que la suma de los desempeños individuales sea igual a un valor de rendimiento del equipo t_e constante. Haciendo uso de las propiedades podemos resolver esta integral de forma analítica, obteniendo como resultado que el desempeño a priori de los equipos es una distribución gaussiana centrada en la suma de las estimaciones medias $\mu_e = \mu_i + \mu_j$ con una varianza que incluye tanto las incertidumbres de las estimaciones, $\sigma_i^2 + \sigma_j^2$, como la varianza de los rendimientos individuales, $\beta^2 + \beta^2$. Tenemos acceso a este mensaje cuando usamos el operador + de la clase `Gaussian` para sumar los desempeños de los agentes,

```
ta = p1 + p2; tb = p3 + p4
```

Code 4: Computando el desempeño a priori de los equipos

El siguiente mensaje, que envía el factor diferencia f_{d_1} a la variable diferencia d_1 es,

$$\begin{aligned} m_{f_{d_1} \rightarrow d_1}(d) &= \iint \delta(d = t_a - t_b) \mathcal{N}(t_a | \mu_a, \sigma_a^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_a dt_b \\ &= \mathcal{N}(d | \underbrace{\mu_a - \mu_b}_{\text{Diferencia esperada : } \psi}, \underbrace{\sigma_a^2 + \sigma_b^2}_{\text{incertidumbre total : } \vartheta^2}) = \mathcal{N}(d | \psi, \vartheta^2) \quad (\mathbf{d} = \mathbf{ta} - \mathbf{tb}) \end{aligned}$$

La diferencia de desempeños a priori es una distribución gaussiana centrada en la diferencia esperada a priori $\psi = \mu_a - \mu_b$ con una varianza $\vartheta^2 = \sigma_a^2 + \sigma_b^2$ que incluye la incertidumbre de ambos equipos. Tenemos acceso a este mensaje cuando usamos el operador `-` de la clase `Gaussian` para obtener la diferencia de los desempeños de los equipos,

```
d = ta - tb
```

Code 5: Computando la diferencia de desempeños a priori

El último mensaje descendente, el que envía el factor f_r a la variable r , permite computar la evidencia, es decir la predicción a priori del resultado observado.

$$m_{f_r \rightarrow r}(r) = \int \mathbb{I}(d > 0) \mathcal{N}(d|\psi, \vartheta^2) dd = 1 - \Phi(0|\psi, \vartheta^2) \quad (\text{evidence})$$

Tenemos acceso a este mensaje calculando el valor acumulado desde 0 hasta ∞ de la distribución de diferencia de desempeños de los equipos.

```
e = 1.0 - cdf(0, d.mu, d.sigma)
```

Code 6: Computando la predicción a priori del resultado observado (o evidencia)

Donde `e` contiene el valor de la ecuación `evidence`.

3.2. Menajes ascendentes

La necesidad de aproximar el posterior de las habilidades, que en su expresión exacta (vista en la teórica), puede resolverse aproximando el posterior de la la diferencia de desempeños, que es una gaussiana truncada.

$$p(d|\text{Gana}) \propto \mathcal{N}(d|\psi, \vartheta^2) \mathbb{I}(d > 0) \quad (1)$$

Se sabe que la familia exponencial, a la que pertenece la distribución gaussiana, minimiza la divergencia Kullback-Leibler respecto de la verdadera distribución p , $KL(p||q)$, cuando ambas tienen mismos momentos [?]. Sea $E(d|d > 0)$ y $V(d|d > 0)$ la esperanza y la varianza de $p(d|\text{Gana})$, luego la gaussiana que mejor aproxima a $p(d|\text{Gana})$ es

$$\hat{p}(d|\text{Gana}) = \mathcal{N}(d|\hat{\psi}, \hat{\vartheta}^2) = \mathcal{N}\left(d | E(d|d > 0), V(d|d > 0)\right) \quad (d > 0)$$

En nuestra clase `Gaussian`, implementamos el método `greater than __gt__(self, threshold)` para que realice automáticamente la aproximación de la gaussiana truncada.

```
d_approx = d > 0
```

Code 7: Computando la aproximación de la diferencia de desempeños

donde la distribución de diferencias `d` fue inicializada en el código 5. Dada $\hat{p}(d|\text{Gana})$, podemos calcular el resto de los mensaje ascendentes usando las operaciones de la clase `Gaussian`. Para derivar el primer mensaje ascendente aproximado, vamos a usar el hecho de que la marginal de la diferencia d y el resultado Gana se puede calcular como el producto de los mensajes entrantes a la variable d en el *factor graph* que tiene al resultado como variable observada.

$$p(d, \text{Gana}) = \underbrace{m_{f_d \rightarrow d}(d)}_{\text{Prior}} \underbrace{m_{f_r \rightarrow d}(d)}_{\text{Likelihood}} = p(d) P(\text{Gana}|d)$$

y que el likelihood de la variable d , $m_{f_r \rightarrow d}(d)$, es igual al mensaje ascendente que envía la variable d al factor f_d , $m_{d \rightarrow f_d}(d)$.

$$\begin{aligned} m_{d \rightarrow f_d}(d) &= m_{f_r \rightarrow d}(d) = \frac{p(d, \text{Gana})}{p(d)} \\ &\propto \frac{p(d|\text{Gana})}{p(d)} \approx \frac{\hat{p}(d|\text{Gana})}{p(d)} \\ &= \frac{\mathcal{N}(d|\hat{\psi}, \hat{\vartheta}^2)}{\mathcal{N}(d|\psi, \vartheta^2)} = \mathcal{N}(d, \psi_{\div}, \vartheta_{\div}^2) \end{aligned} \quad (1h.d)$$

El operador división / implementado en el métodos `__div__(self, M)` de nuestra clase `Gaussian` devuelve la gaussiana $N(d, \psi_{\div}, \vartheta_{\div}^2)$, que es el likelihood aproximado de la variable d .

```
lh_d = d_approx / d
```

Code 8: Computando el primer mensaje aproximado

Todos estos mensajes aproximados pueden ser interpretados como verosimilitudes porque contienen la información del resultado observado. El mensaje aproximado que el factor diferencia f_d envía a la variables de desempeño del equipo ganador t_a es,

$$\begin{aligned}\hat{m}_{f_d \rightarrow t_a}(t_a) &= \iint \underbrace{\delta(d = t_a - t_b)}_{t_a = d + t_b} \mathcal{N}(d_1 | \psi_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dd dt_b \\ &= \int \mathcal{N}(t_a - t_b | \psi_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b = \mathcal{N}(t_a | \mu_b + \psi_{\div}, \vartheta_{\div}^2 + \sigma_b^2)\end{aligned}\quad (\text{lh_ta})$$

Lo que se resume en nuestro operador +.

```
lh_ta = lh_d + tb
```

Code 9: Likelihood aproximado del desempeño del equipo a , t_a .

El mensaje aproximado que el factor desempeño de equipo f_{t_a} envía a la variables de desempeño individual p_1 es,

$$\hat{m}_{f_{t_a} \rightarrow p_1}(p_1) = \iint \underbrace{\delta(t_a = p_1 + p_2)}_{p_1 = t_a - p_2} \mathcal{N}(t_a | \mu_b + \psi_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta^2) dt_a dp_2 \quad (\text{lh_p1})$$

Lo que se resume en nuestro operador -.

```
lh_p1 = lh_ta - p2
```

Code 10: Likelihood aproximado del desempeño del jugador 1, p_1 .

El mensaje aproximado que el factor de desempeño individual f_{p_1} envía a la variables de habilidad s_1 es,

$$\begin{aligned}\hat{m}_{f_{p_1} \rightarrow s_1}(s_1) &= \int \mathcal{N}(p_1 | s_1, \beta^2) \mathcal{N}(p_1 | \mu_1 - \psi + \psi_{\div}, \vartheta_{\div}^2 + \vartheta^2 - \sigma_1^2 - \beta^2) dp_1 \\ \hat{m}_{f_{p_1} \rightarrow s_1}(s_1) &= \iint \underbrace{\delta(p_1 = s_1 + o)}_{s_1 = p_1 - o} \mathcal{N}(o | 0, \beta^2) \mathcal{N}(p_1 | \mu_1 - \psi + \psi_{\div}, \vartheta_{\div}^2 + \vartheta^2 - \sigma_1^2 - \beta^2) dp do \quad (\text{lh_s}) \\ &= \mathcal{N}(s_1 | \mu_1 - \psi + \psi_{\div}, \vartheta_{\div}^2 + \vartheta^2 - \sigma_1^2)\end{aligned}$$

Donde o representa el ruido descrito en el segundo mensaje descendente.

```
lh_s1 = lh_p1 - o
```

Code 11: Likelihood aproximado de la habilidad.

Finalmente, el posterior proporcional aproximado de la variable s_1 se obtiene multiplicando los mensajes que recibe de sus factores vecinos.

$$\hat{p}(s_1, r) = \mathcal{N}(s_1 | \mu_1, \sigma_1^2) \mathcal{N}(s_1 | \mu_1 - \psi + \psi_{\div}, \vartheta_{\div}^2 + \vartheta^2 - \sigma_1^2) \quad (\text{posterior})$$

Tenemos acceso al posterior normalizado usando el operador * de la clase `Gaussian`.

```
posterior_s1 = s1 * lh_s1
print(posterior_s1)
> N(mu=2.361, sigma=5.516)
```

Code 12: Computando el posterior aproximado