



Matias Mazzanti

Estimación de habilidad y la asignación de *handicap*. Aplicación al juego de go

Tesis de Licenciatura en Ciencias Físicas
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Enero 2020

Director: Esteban Mocskos

Co-director: Pablo Balenzuela

TEMA:

ALUMNO: Matias Mazzanti

L.U. N°: 110/13

LUGAR DE TRABAJO: Laboratorio Interdisciplinario de Computación de Alto Rendimiento, Departamento de Computación, FCEyN, UBA.

DIRECTOR DEL TRABAJO: Dr. Esteban Mocskos

CODIRECTOR o COLABORADOR: Dr. Pablo Balenzuela

FECHA DE INICIACIÓN: Octubre 2019

FECHA DE FINALIZACIÓN: Diciembre 2020

FECHA DE EXAMEN:

INFORME FINAL APROBADO POR:

Autor: Matias Mazzanti

Jurado: Santiago Laplagne

Director: Esteban Mocskos

Jurado: Diego Shalom

Profesora: Silvina Ponce Dawson

Jurado: Juan Kamienkowski



Resumen

En diversas áreas, tales como los sistemas educativos, los deportes y la industria de los videojuegos, resulta crucial el establecimiento de *rankings* en los que se clasifican a quienes practican determinada actividad. La habilidad, la cual cuantifica de forma relativa a la comunidad la eficiencia de un agente en realizar una dada actividad, es una variable oculta, por lo que solo se la puede estimar a partir de sus consecuencias observables directas: el producto de resolución de problemas y competencias. Dentro de las diversas prácticas deportivas y de entretenimiento, es habitual la asignación de ventaja (i.e. *handicap*) para que un juego sea desafiante y, a la vez, accesible para los jugadores. Un ejemplo clásico es el juego de mesa go, en el cual a los participantes que tienen una diferencia de habilidad considerable, se les asigna una cantidad de fichas extra al principio de la partida, con el fin de lograr un encuentro interesante. Los modelos para estimar habilidad se basan en que la probabilidad de un resultado observado depende de la diferencia de rendimiento entre los agentes que compiten. El rendimiento depende de una cantidad indeterminada de factores desconocidos que pueden afectar a cada competidor en un momento dado. TrueSkill es un método que utiliza la inferencia bayesiana para poder realizar una estimación de las habilidades relativas de los agentes en un sistema basado en la observación de los resultados entre todos ellos. En la presente tesis, se propone un método basado en TrueSkill para poder analizar la forma de asignación de *handicap* en el juego de go en base a información de partidas recolectadas del servicio en línea KGS.

La propuesta se basa en tratar el *handicap* como un jugador adicional que hace equipo con el jugador más débil. Gracias a esto, se logra estimar no solo las habilidades de los jugadores involucrados sino, a su vez, el valor que aportan las fichas de ventaja. Ésto permite definir una nueva forma de asignación de *handicap* en las partidas de go. En este trabajo se compara nuestra propuesta de asignación de *handicap* con la tradicional y se muestra que con nuestra propuesta se lograrían partidas más parejas. Se analizaron las partidas en las cuales fue asignado *handicap* y se vio que en un 5 % no era requerido y en un 91 % el valor dado estuvo mal asignado. A su vez, se analizaron las partidas en las cuales no fue asignada ninguna ventaja y se vio que en un 57 % hubiera sido requerido. Se concluye que el método desarrollado permitiría una mejor asignación de *handicap* en el juego de go, pero también abre la posibilidad para su aplicación en otras actividades.

Keywords: Habilidad, go, handicap, inferencia bayesiana

Índice general

1	Introducción	1
1.1	Motivación	1
1.1.1	Go	6
1.1.2	TrueSkill	7
1.1.3	TrueSkill Through Time	9
2	Fundamentos	10
2.1	Introducción teórica	10
2.1.1	Modelo probabilístico	10
2.1.2	Distribución gaussiana	12
2.1.3	Gráfico de factores	14
2.1.4	Pasaje de mensajes	15
2.1.5	Expectation Propagation	17
2.1.6	Metodología de evaluación	18
2.2	Modelos	19
2.2.1	Elo	19
2.2.2	TrueSkill	22
2.2.3	TrueSkill insights	25
2.2.4	Full TrueSkill	35
2.2.5	TrueSkill Through Time	37
2.2.6	Visualización	39
3	Validación	41
3.1	Validación	41
3.2	Análisis del modelo	44
4	Resultados	50
4.1	Data set	50
4.1.1	Ranking go	52
4.1.2	Caracterización	53
4.1.3	Curvas de aprendizaje	55
4.2	Análisis Handicap	56
4.2.1	Implementación	58
4.2.2	Distribución inicial y final	69
4.2.3	Análisis predictivo	70
5	Conclusiones y trabajo a futuro	73
A	Propiedades de las gaussianas	78
A.1	Propiedades	78
A.2	Suma de n normales	78
A.3	Multiplicación normales	80

A.3.1	Dos gaussianas	80
A.3.2	Dos gaussianas de misma variable integradas	81
B	Métodos	83
B.1	TrueSkill	83
B.1.1	Mensajes descendentes	83
B.1.2	Ascendentes	86
B.1.3	Posterior ganador y perdedor	90
B.2	Ejemplo 1 vs 1-Handicap	99
B.2.1	Full TrueSkill	103
B.2.2	TrueSkill Through Time	105

Capítulo 1

Introducción

1.1 Motivación

A lo largo de nuestra historia humana, una de las características que siempre resaltó en nuestra especie, a diferencia de otras, es la forma y facilidad que tenemos a la hora de aprender a realizar diversas tareas. Esta facilidad permite a una persona, poder adquirir la capacidad de realizar tareas de diversa índole. Esto no necesariamente significa que toda persona que aprende una determinada tarea, la realice con la misma efectividad que el resto. A su vez, tampoco significa que para adquirir dicho conocimiento y obtener una determinada eficiencia en su realización, todos necesiten la misma cantidad de tiempo y práctica. Cada persona tendrá diferencias a la hora del aprendizaje y a qué nivel final de eficiencia llegue [1–4].

Un ejemplo claro de la extraordinaria manera de aprendizaje y del grado de complejidad queda representada por la adaptabilidad del ser humano. Durante mucho tiempo se creyó que el éxito adaptativo, aprendizaje a la supervivencia en un entorno, se debía únicamente a nuestras capacidades cognitivas individuales. Sin embargo, nuestra biología es la misma que la de nuestros ancestros cazadores-recolectores. Lo que cambió radicalmente en estos más de cien mil años no fue nuestra información genética, sino nuestra información cultural. A diferencia de otros animales, los humanos tenemos una capacidad extraordinaria de imitar [5]. Cada uno de nuestros ancestros imitó, modificó y transmitió parte de su conocimiento a la siguiente generación. El **sistema de información** cultural heredado (herramientas, creencias y prácticas), no solo permite a las nuevas generaciones evitar tener que adquirirlas por experiencia individual, sino que permite alcanzar niveles de conocimiento que jamás podrían ser reproducidos por un individuo durante su período de vida. Es en la supervivencia de los sistemas culturales que se dan las condiciones para el desarrollo de conocimiento altamente especializado [6]. De esta forma, el **agente** de conocimiento dejan de ser los individuos y pasan a ser los sistemas sociales, portadores efectivos de la información trans-generacional. La información acumulada en una sociedad, como lo es una comunidad de juegos en línea, se denomina **cultura**.

Otra de nuestras características como humanos que viene asociada es la de enseñar y, por ende, aprender entre nosotros siendo capaces de aumentar la eficiencia en todo tipo de actividad. Esta clase de comportamiento se da tanto en ejemplos de la actualidad como durante toda la historia. Hoy en día se presenta en cualquier tipo de competencia deportiva, en donde se pretende que un agente realice una tarea bajo ciertas reglas pautadas, como puede ser el tenis, de forma tal que se maximice la eficiencia en dicha actividad.

Sin importar en que época de la historia se vea, el humano siempre buscó la forma

de categorizar las diferentes actividades para diferenciar a los individuos de su sociedad. Esta categorización se utiliza de manera explícita o implícita, tanto en la toma de un empleado buscando contratar personal con los requisitos necesarios para el puesto, como en el intento de predecir el ganador de una competencia deportiva. El seguimiento de esta cuantificación en el tiempo resulta útil y de suma importancia, debido a que se puede extrapolar información de diversas índoles tanto del agente como su entorno. Por ejemplo, las sociedades deportivas tienen como objetivo que sus eventos competitivos estén determinados por la habilidad de sus participantes y no por factores externos prohibidos. Esto puede ser desde el caso del dopaje como también en la utilización de equipamiento prohibido por las reglas de dicho deporte. Un ejemplo claro es el automovilismo, en donde cada vehículo tiene que cumplir determinadas características y condiciones técnicas estipuladas para que, de esta forma, la diferencia recaiga más fuertemente en la capacidad del piloto y no tanto en el automóvil.

Tener bien caracterizada la evolución de la habilidad en la realización de una tarea, tanto deportiva como productiva, puede servir para buscar anomalías o factores que impacten en el desarrollo de una persona. Mientras más precisa sea esta categorización más útil será la información que se pueda extraer. Esta noción de cuantificación acerca de qué tan buena o qué tan apta es una persona (o agente) en comparación a otros para realizar una actividad es lo que se denomina **habilidad**. La habilidad representa la capacidad de una persona para llevar a cabo correctamente y con facilidad una determinada actividad, trabajo u oficio. Esta propiedad afecta y es analizada en la mayor parte de nuestra sociedad moderna, en todos los deportes, en todos los trabajos, incluso en nuestras casas a la hora de realizarnos la comida. Una forma de representación o visualización de la evolución de la habilidad es la **curva de aprendizaje**, que grafica el cambio de la habilidad en función de los distintos eventos en los que el agente ha participado en un periodo de tiempo tal como se presenta en la figura 1.1. La habilidad (medida en el eje vertical) generalmente aumenta con el aumento de la experiencia (el eje horizontal), es decir, cuanto más realiza una tarea, mejor se vuelve en ella. Uno de los modelos mas utilizados para la descripción de dicha curva es mediante una función sigmoide, la cual divide el aprendizaje en tres etapas bien definidas: un primer estudio en el cual la persona experimenta, pero su habilidad no mejora notablemente (*Stage I*). En cierto momento, el agente comienza una segunda etapa de mejora en su habilidad caracterizado por un fuerte cambio en su habilidad (*Stage II*). Finalmente el agente alcanza su máximo nivel y, a partir de aquí, de existir mejoras resultan marginales (*Stage III*). De esta forma, teniendo la curva de aprendizaje de los agentes se puede por ejemplo analizar en que etapa se encuentran y proyectar de forma estimativa su evolución.

Una de las dificultades principales radica en cómo realizar la cuantificación de la habilidad de una persona o agente. Esta dificultad se basa en que la habilidad no se puede observar directamente, sino que se observa el resultado de un determinado evento en un momento dado. Por ejemplo, un jugador de tenis puede jugar un partido muy bueno un día y ganar, pero al siguiente tener una mala tarde y perder contra el mismo contrincante sin que ninguno de los dos jugadores hayan cambiado, esencialmente, su habilidad. Esto introduce el concepto de **rendimiento**: dada una habilidad, el rendimiento representa qué tan bien se ejecutó en un evento determinado. Esto es de importancia ya que el resultado que obtiene un agente con una dada habilidad en un evento particular, está condicionado a múltiples factores. Como ya se comentó, el rendimiento depende de la habilidad, esto se puede ver de manera directa en el ejemplo del tenista, en el cual podemos entender que el rendimiento de un jugador profesional siempre será mayor al de alguien que desconoce o es inexperto en ese deporte debido a la diferencia extrema de habilidad. A su vez, el rendimiento puede depender de diversos factores, como puede ser el clima ambiental en el

día del partido, tipo de cancha a jugar (polvo de ladrillo, cemento, etc), la salud del jugador, la indumentaria o accesorios utilizados, entre otros. Podemos concluir que el rendimiento está relacionado a la habilidad de forma **multi-causal**. Como ya mencionamos, la habilidad es una variable oculta que no puede ser medida directamente, lo que se busca es poder estimarla mediante observaciones directas de los resultados finales en eventos puntuales en los cuales podemos observar de manera cuantitativa los rendimientos de los agentes involucrados. Es decir, quién ganó o perdió, quién le fue mejor o peor, quién tuvo mayor o menor rendimiento en un evento.

Fuera del área deportiva, nosotros podríamos probar una misma receta preparada por dos cocineros diferentes e indicar cuál nos pareció más rica, es decir quién realizó de mejor forma una determinada actividad y, de ese modo, asignar cuál de ellos tuvo mejor rendimiento al momento de realizar la receta pedida. Pero, considerando un único evento, no podemos decir cuál de los dos es “mejor” cocinero, es decir cuál tiene mayor habilidad en la cocina. Otro ejemplo resulta en comprobar los conocimientos en matemática de los alumnos de una escuela. Usualmente, esto se logra mediante un examen y se analiza quién sacó mayor puntuación, pero nuevamente, esto solo representa el rendimiento en ese examen particular, en ese evento determinado. En estos dos ejemplos, solo se estaría registrando el rendimiento en un evento particular, resulta en una pista para saber cuál es la habilidad pero no es la habilidad directamente. Podríamos generalizar diciendo que lo que nos interesa es utilizar eventos puntuales para obtener un conocimiento global acerca de la habilidad del agente. Es decir, nos interesa saber qué tanto sabe de matemática un alumno y no si puede resolver un ejercicio puntual. El problema radica en que a pesar de que el concepto de habilidad nos es familiar e intuitivo, no existe una forma universal de cuantificar de manera eficiente este atributo oculto. El mayor inconveniente reviste en la forma de evaluar para poder estimar la habilidad. Puede suceder que un cocinero sea mucho mejor para cocinar justo el plato que probamos (por ejemplo, accidentalmente derramó un condimento durante la preparación de la receta y esto le otorgó un sabor exquisito), pero no así para el resto de las recetas, obteniendo posiblemente menor habilidad global que el otro cocinero.

Hemos mencionado qué tan bueno o malo, quién es mejor o peor, pero no obtuvimos un valor numérico que pueda describir la habilidad de un agente en una actividad. La manera ideal de representar la habilidad todavía no es conocida, siendo aún un problema abierto, aunque se han realizado diferentes avances en este sentido [7–9].

Resulta de suma importancia poder obtener una magnitud de habilidad que podamos utilizar para comparar en una determinada actividad a diferentes personas/agentes dentro de una población, también resulta de mucho interés poder evaluar su evolución en el tiempo. Es importante entender que nos interesa lograr estimar la habilidad en escala relativa, ya que el valor absoluto individual no aporta información relevante. De esta manera, se podría clasificar a una población de agentes en una cierta actividad obteniendo un *ranking* basado en la habilidad de los agentes. Con esta información, por ejemplo, se podría contratar de manera eficiente a empleados en un determinado puesto tal que cumplan los requisitos buscados, como también poder elegir aquellos agentes que sean más eficientes en la tarea buscada.

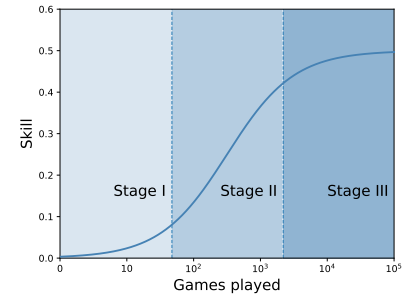


Figura 1.1: Idealización de la evolución del cambio en la habilidad de un jugador respecto a su experiencia. Se resaltan las tres etapas de una curva de aprendizaje típica.

Otra aplicación de importancia es poder hacer un seguimiento de la evolución de la habilidad en el tiempo. Esto abre la puerta a estudiar el aprendizaje de los diferentes agentes de una población. Un ejemplo de interés social, dentro del área de educación, es el de analizar la evolución de la habilidad de los alumnos en las diferentes materias. De esta manera, se podrían comparar diferentes métodos de aprendizaje en las escuelas, observando cuál de éstos logró aumentar más la habilidad media de los alumnos. Otro caso podría ser el estudio de la efectividad de distintos planes de entrenamiento en los deportes, tratando de establecer con cuál se obtienen los mejores resultados. Es decir, podemos observar cómo influyen ciertos factores externos en el rendimiento de un individuo en una determinada actividad. Por ejemplo, se podría analizar el desempeño de tenistas respecto a la utilización de una nueva tecnología de encordado de la raqueta. Al tener información del resultado de las partidas a partir de la adopción del nuevo encordado, se podría determinar su impacto. Es decir, este tipo de análisis requiere una cantidad de información previa del agente y su entorno para lograr extraer conclusiones. Mientras más partidas tengamos de una persona y su entorno, más certera será nuestra estimación de su habilidad.

Como podemos notar, el poder estimar la habilidad de un agente no es una tarea trivial. Existen muchos factores diferentes en cada tipo de acción, dificultando su modelado. A lo largo de los años se crearon diferentes métodos para diferentes actividades. En general los modelos toman en cuenta el rendimiento. Uno de los primeros métodos que se popularizó a partir del ajedrez es el de **Elo**, presentado en 1960 [8, 10]. Surgió ante la necesidad de mejorar el sistema de ranking del ajedrez, aunque su uso se amplió a diversos juegos. El modelo supone a la habilidad como un escalar. A su vez, define al rendimiento como el valor de habilidad que realmente se tiene en un dado evento, modelando así la posibilidad de que un mismo jugador se desempeñe de diferentes formas en distintos eventos. El rendimiento se modela mediante una distribución de probabilidad con forma de campana centrada en la habilidad, y con una desviación estándar fija para todo jugador dentro del mismo deporte o tarea. El método supone que el jugador con mayor rendimiento es el ganador. De esta forma, observando el resultado en un juego dado, el método de **Elo** actualiza la distribución de habilidad usando la diferencia de habilidad entre los contrincantes. Si la diferencia de habilidad es grande y el ganador es el jugador de menor habilidad, el ajuste de habilidad posterior debería ser grande ya que ocurrió un resultado no esperado (i.e. la *sorpres*a del resultado es alta), aumentando la estimación de habilidad del ganador y disminuyendo la de del perdedor. Por el otro lado, si el resultado diera como ganador al jugador de mayor habilidad, el cambio posterior en la habilidad debería ser mucho más leve ya que ocurrió lo que se suponía que podía pasar (i.e. no hay sorpresa en el resultado).

Mientras más información tengamos de un jugador (mayor número de partidas), mayor certeza tendremos acerca de su habilidad. En los casos de los deportes y juegos, poder tener una medida de habilidad de los participantes tiene una gran importancia para poder generar mejores experiencias de juegos, más competitivas y desafiantes. Un partido o una competencia con jugadores o equipos de habilidades similares, resultaría en que los participantes tengan probabilidades parejas de ganar, manteniendo así el interés en la actividad propuesta.

Gracias a la revolución de internet y a la creciente accesibilidad de datos, los juegos en línea resultan una de las más interesantes fuentes de información, que incluyen en algunos casos, grandes bases de datos de fácil acceso. Tal es el caso del ajedrez, el cual cuenta con múltiples servidores en línea en los cuales jugadores de todo el mundo juegan entre sí. Cada servidor utiliza alguna forma de ranking, en general basada en el ya mencionado método de *Elo*. Estas partidas se almacenan en bases de datos internas pudiendo superar las millones de partidas. De manera similar se encuentra el milenar juego go que utiliza

actualmente un sistema de ranking basado en *Elo* y cuenta también con grandes bases de datos de partidos. El go es un juego de estrategia de un jugador en contra de otro, es de origen chino y se remonta a más de 2500 años atrás. Se juega en un tablero cuadrulado que puede tener diversos tamaños, aumentando la dificultad del juego mientras éste sea más grande. Se usan fichas, que representan piedras del tamaño de una moneda, de color blanco y negro para cada uno de los jugadores. Los puntos se cuentan por la cantidad de fichas capturadas del rival así como por la cantidad de tablero dominado por cada jugador. Al finalizar la partida, el jugador con mayor cantidad de puntos es el ganador.

Este juego dispone de diversos portales en línea, contando con múltiples servidores gratuitos de go en los cuales cientos de jugadores se enfrentan y toman millones y millones de decisiones por hora, convirtiéndolo en un experimento cognitivo de índole masiva. Una de las cuestiones interesantes de este juego se basa en la simplicidad de sus reglas, mucho más sencillas que las del ajedrez. Sin embargo, su complejidad es aún mayor que éste. En la actualidad, la solución por fuerza bruta sigue fuera del alcance computacional. Además, el juego de go tiene algunos problemas propios que todavía no han sido resueltos. Por ejemplo, se suele compensar a los jugadores de menor habilidad por medio de una ventaja (*handicap*) de manera de emparejar la partida cuando se enfrenta a un rival mejor. A pesar de existir reglas históricas para la asignación de *handicap*, no está eficientemente resuelto cómo impacta la utilización de *handicap* debido a que los jugadores podrían presentar una gran diferencia de habilidad entre sí. El método actual no contempla al *handicap* ya que no tiene caracterizado y cuantificado el impacto de éste en el juego. Encontrar una buena estimación en estas condiciones, permitiría no solo mejorar la predicción sobre la habilidad, sino también tener una mejor escala para la asignación de *handicap* en las partidas.

En esta tesis, se eligió el método de estimación de habilidad **TrueSkill** como herramienta para la estimación de habilidad. Este método resuelve algunas de las limitantes de *Elo*. Una limitación de importancia en *Elo* es que la actualización de habilidad se realiza manera simétrica entre ambos jugadores: lo que gana un jugador en habilidad, es lo que pierde el otro. Esto produce que las estimaciones para un nuevo jugador (al cual se le asigna una habilidad inicial fija) puedan generar grandes cambios cuando éste compite con un jugador cuya habilidad tenía un buen grado de certeza. A pesar de que se realizaron modificaciones para superar estas limitaciones, éstas se basan en soluciones ad-hoc, que no se pueden generalizar. **TrueSkill** soluciona esto usando un marco probabilístico, como se verá más adelante. Además, éste permite estimar eventos que involucran a más de dos jugadores e, incluso, a múltiples equipos de varios jugadores. Otra de las mejoras introducidas es la posibilidad de modelar eventos con posibilidad de empate, es decir, ni una victoria ni una derrota.

En 2007 Dangauthier et al. [11] propuso una mejora a **TrueSkill** denominada **TrueSkill Through Time** (TTT). Una de las principales diferencias es que la información de la habilidad no se propaga en un único sentido temporal (del pasado al futuro), si no que éste busca la convergencia, propagando esta información también en sentido contrario. Más detalles de este método se presentan más adelante en ese capítulo y, con mayor detalle, en la sección 2.2.5.

Esta tesis se basa en aplicar técnicas de estimación de habilidad para proponer una alternativa a la asignación de *handicap*. En particular, se hace foco en el juego de mesa go y se utiliza un método basado en **TrueSkill** para el análisis de la habilidad de los jugadores. Se busca obtener una propuesta metodológica para la asignación de *handicap* que genere partidos más parejos y, a la vez, desafiantes.

1.1.1 Go

El go es un juego de mesa de estrategia para dos personas. Sus orígenes giran en torno a la historia de la antigua china, pues se estima que fue originado hace 2500 a 4000 años atrás. Es el juego más antiguo que aún se juega en su forma original. Consiste en la colocación de fichas en las intersecciones de un tablero cuadrículado. La sencillez de este, presentaba en la antigüedad, la posibilidad de poder marcar una grilla en la tierra y con unas simples piedras, poder jugar. En la actualidad, existen diferentes tableros del juego dependiendo a su tamaño, entre los más comunes 9×9 , 13×13 y 19×19 , siendo este último el más usual. En la figura 1.2 se muestra un ejemplo correspondiente al primer caso.

Como podemos observar, se cuenta con dos colores de fichas, blancas y negras, siendo un color por jugador. A diferencia del ajedrez, el jugador con fichas negras comienza el juego, las piedras o fichas de un jugador son indistinguibles entre sí y, una vez jugadas, no se pueden volver a mover. Se puede capturar una piedra o un conjunto de piedras y eliminarlas del tablero si están completamente rodeadas por piedras de otro color. Uno de los objetivos principales es la de formar territorios. Estos son zonas del tablero en el cual están dominados por un jugador, sin posibilidad de perderlos ante el rival. De modo resumido, al final de la partida, gana el jugador con mayor territorio y fichas capturadas.

A pesar de tener reglas sencillas, el go presenta una complejidad muy alta debido a la diversidad de jugadas posibles en cada movimiento, requiriendo de una estrategia compleja. El juego consiste en acumular puntos al final de la partida, mediante la captura de piedras ajenas y de territorio. En el juego de go, existen tres formas de victoria:

- I) El jugador que obtenga mayor cantidad de puntos al finalizar la partida, es el ganador.
- II) La segunda forma, y quizás la más obvia, es por el abandono de uno de los dos jugadores. En general esto sucede, cuando uno de los dos jugadores toma noción de que no puede ganar la partida por los puntos acumulados por el otro jugador.
- III) La última forma consiste en perder por falta de tiempo. De manera similar al ajedrez, cada jugador cuenta con determinado tiempo que se va consumiendo en cada turno de dicho jugador.

En un juego de go, el jugador con fichas negras (jugador negro), es el que comienza el juego, poniendo la primera ficha en el tablero. Como el hecho de iniciar la partida permite estar siempre una jugada adelante que el jugador con fichas blancas (jugador blanco), existe una regla llamada regla del **Komi** que, en la mayoría de veces, le asigna al jugador blanco un número de puntos extra para el conteo final buscando contrarrestar la ventaja que tiene el jugador negro. Estos puntos no tienen ninguna influencia en el juego hasta que éste haya terminado y se cuenten los puntos totales. Los puntos por *komi* no tienen un valor aceptado a nivel mundial, teniendo cada zona un diferente valor. A su vez, como se suele jugar sin la posibilidad de empate, para evitarlo se usa un valor de *komi* no entero. El valor de *komi* más utilizado es el de 6,5. En muchos otros casos, se utiliza el *komi* de 0,5 que simplemente evita el empate, pero no asigna una ventaja al blanco por el hecho de jugar en segundo lugar.

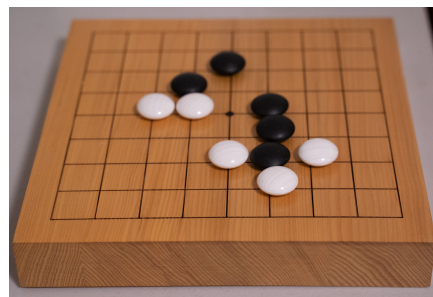


Figura 1.2: Tablero de go de dimensiones 9×9 . Se muestra un juego, luego de cinco movimientos por jugador.

Este juego posee un sistema de clasificación de los jugadores muy caracterizado, teniendo un sistema de rangos (jerarquía) de *Kyu* (la inferior) y *Dan* (la superior) similar al sistema característico de las artes marciales. Estos niveles van del *Kyu* 30 al *Kyu* 1 y, luego, del *Dan* 1 al *Dan* 9 ordenados de principiante a maestro.

El go, tiene un sistema para poder equilibrar las partidas cuando la diferencia de rango entre los dos jugadores es muy grande. Este es el concepto de **handicap** que corresponde con dar ventaja a un jugador sobre otro. En general, se asigna con las fichas negras al jugador de menor rango (habilidad) y, dependiendo la diferencia de rango, este jugador inicia el juego con una cantidad adicional de fichas en el tablero. Es decir, si un jugador en una determinada partida tiene un *handicap* de 2, esto representa que, además de iniciar la partida con una ficha, éste tiene una ficha adicional en el primer movimiento. Si tuviera *handicap* 5, además de su primer ficha en su primer movimiento, éste tendría un adicional de cuatro fichas, en esta primera instancia. En general, estos casos se suele utilizar el valor de *komi* de 0,5 y, además, estas fichas de *handicap* tienen una posición asignada fija.

Tanto el *komi* como el *handicap*, son conceptos independientes. Uno representa la ventaja al jugador blanco por iniciar segundo la partida, como también imposibilitar el empate. El otro representa la ventaja al jugador de menor habilidad. Como ya se mencionó, el problema sigue estando en encontrar una escala de valores, en el cual se pueda encontrar el valor ideal para el *komi*. Además esta escala tiene que poder representar al *handicap* de manera tal que, dados dos jugadores con diferente valor de habilidad, se pueda asignar la ventaja que vuelva a la partida lo más pareja posible.

1.1.2 TrueSkill

En el marco de las comunidades de juegos en línea, así como en las áreas de la educación y deportes, una de las características principales es el concepto de **habilidad**. Como ya fue mencionado, la habilidad corresponde con cuantificación de la noción de qué tan bueno o apto es un agente en realizar una actividad en particular, en general en comparación a otros. En el contexto de juegos en línea, la habilidad representa qué tan bien juega a un determinado juego un agente de la comunidad en relación al resto de los miembros de una comunidad.

Una de las dificultades radica en que la habilidad es una variable que generalmente resulta imposible de medir directamente, es una variable oculta. En la práctica, lo mejor que podemos hacer es estimarla a partir de sus consecuencias observables más directas. Una solución ingenua podría ser el observar la frecuencia de resultados positivos (victorias de un jugador), pero esto no es un buen indicador debido a que, fundamentalmente, depende también de la dificultad de los desafíos (habilidad del contrincante). Por esa razón, todos los modelos convencionales se basan en *pairwise comparisons*. Ya los primeros modelos generativos, propuestos hace casi un siglo [12, 13], suponen que la probabilidad de un resultado observado, r_{ij} , entre un jugador i en contra de un agente j es $p(r_{ij} | d_{ij})$ la cual depende de la diferencia de rendimiento, d_{ij} , entre el agente i y su contrincante j . El campo se revitalizó 20 años después gracias a las publicaciones de Bradley y Terry [14] y Moteller [15]. Actualmente, los estimadores de habilidad más utilizados son los propuestos por [9, 10, 16] y algunas de las variantes de la *Item-Response Theory* recopiladas por [17, 18]. Un gran avance produjo la metodología desarrollada por [10] para la Federación de Ajedrez de los Estados Unidos (USCF), adoptada hasta el día de hoy por la Federación Internacional de Ajedrez (FIDE).

La figura 1.3 ofrece una representación gráfica en la que las habilidades ocultas generan

los resultados observables. La relación causal no es directa. Los agentes tienen una habilidad específica en cada momento (s_i y s_j) y el desempeño (p_i y p_j) que exhiben depende de una cantidad indeterminada de factores desconocidos que se modelan como un ruido aleatorio gaussiano alrededor de la media de habilidad (s_i y s_j) expresado como $\mathcal{N}(p | s, \beta^2)$ para cada uno de los agentes. Según este modelo, los resultados son una consecuencia directa de la diferencia de esos rendimientos ocultos, $d_{ij} = p_i - p_j$. Un agente obtiene un resultado positivo cuando supera el rendimiento de su adversario, $r_{ij} = (d_{ij} > 0)$. Al observar un resultado podemos deducir, dadas las hipótesis, el orden de los rendimientos ocultos que, en efecto, ocurrió en ese evento, y su probabilidad a priori.

La solución metodológica de **Elo** es extremadamente sencilla y astuta: actualizar la estimación actual de habilidad en función de la sorpresa del resultado, actualizando más fuertemente la estimación cuanto mayor es la diferencia entre los jugadores, favoreciendo el caso en el que gana el menos probable. Esta solución puede recuperar la escala relativa de los agentes, partiendo de valores iniciales arbitrarios. Sin embargo, tiene algunas debilidades importantes. Debido a la falta de una noción de incertidumbre sobre las estimaciones, lo que un agente pierde en habilidad lo adquiere el otro. Como las estimaciones sobre nuevos jugadores (poco conocidos al momento) tienden a generar alta sorpresa, su presencia puede modificar las estimaciones bruscamente. Una solución ad-hoc fue propuesta para romper esta simetría del modelo de **Elo**, que consiste en reducir el impacto de la actualización mediante la multiplicación de una constante en función de las veces que el agente ya fue visto por el sistema.

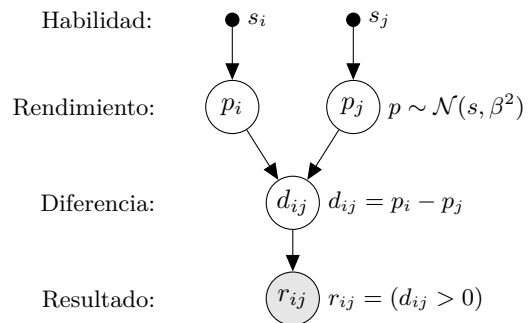


Figura 1.3: Las habilidades ocultas causan los resultados observables a través de la diferencia de los rendimientos de cada jugador.

La Teoría de la Probabilidad nos proporciona un marco consistente para cuantificar la incertidumbre de nuestras creencias [19], posibilitando una opción para no agregar a nuestro modelo soluciones ad-hoc. Al proceso de computar distribuciones de probabilidad luego de observar el valor de alguna variable aleatoria se llama **inferencia**. El uso de la teoría de la probabilidad para manejar la incertidumbre, conocida como inferencia bayesiana, está hoy adoptada de manera generalizada en el ámbito de la inteligencia artificial y el aprendizaje automático fundamentalmente porque ha mostrado ser un instrumento sumamente útil en la práctica.

En el 2006, Herbrich [9] propuso un método probabilístico, denominado **TrueSkill**, en el cual solucionaba varias de las limitaciones de **Elo**. A diferencia de **Elo**, que considera a la habilidad como un único escalar, éste la modela con una distribución gaussiana, teniendo así la media de la distribución como la habilidad más probable, y a la desviación estándar como la incerteza que se tiene. A su vez, al igual que **Elo**, supone un rendimiento p , correspondiente con un muestreo alrededor de la habilidad del jugador, resultando ganador aquel que tuvo mayor rendimiento. De esta forma, observando el resultado en un juego dado, el método de **TrueSkill** actualiza la distribución de habilidad. Mientras más información tengamos de un jugador (mayor número de partidas), mayor certeza tendremos de su habilidad. Es decir, resulta imprescindible contar con un gran volumen de información con el fin de reducir nuestra incertidumbre en la estimación de la habilidad de cada jugador.

1.1.3 TrueSkill Through Time

El algoritmo de **TrueSkill**, presenta algunas limitaciones importantes. Por un lado, la inferencia que se realiza en un determinado rango temporal t depende del orden en el que ocurrieron los eventos.

Un ejemplo sencillo y conceptualmente importante, es el caso de tres jugadores donde cada uno gana y pierde una partida entre ellos. Supongamos que tenemos únicamente tres jugadores: **a**, **b** y **c**, y que en un mismo día el jugador **a** le gana una partida al jugador **b**, luego **b** le gana una partida al jugador **c** y, por último, **c** le gana una partida al jugador **a**. Podemos intuir que la respuesta que buscamos, es que si no teníamos información previa de ningún jugador, cada uno ganó y perdió un juego y, por ende, deberían finalizar con la misma habilidad estimada. En **TrueSkill**, al importar el orden en que toma los eventos, luego de la primer partida supondrá que el jugador **a** es superior a **b**, asignándole a este último menor habilidad. Luego **b** jugaría contra **c** quien, al perder, hará suponer que este último es aún peor. Por último, en la tercer partida se tendría el caso en que un jugador que se supone muy malo (**c**) le gana a un jugador que se supone muy bueno (**a**). **TrueSkill** terminará de actualizar las habilidades estimándolas incorrectamente, en lugar de obtener las tres habilidades iguales, obtiene tres habilidades estimadas diferentes. Dentro de un periodo suficientemente pequeño de tiempo, los resultados de la inferencia deberían ser independientes del orden de los eventos dentro de ese rango temporal.

Por otro lado, otra limitación por parte de **TrueSkill** y su predecesor **Elo** proviene de la cantidad de partidas necesarias para lograr una caracterización satisfactoria de un agente. Una vez estimado un evento la información extraída se utiliza para el siguiente evento sin posibilidad de reutilizarse. Es decir, la información solo se propaga temporalmente en un sentido. Un ejemplo sencillo, es el caso en donde tenemos dos jugadores nuevos para el modelo, **a** y **b**, y un jugador **c** con gran habilidad bien caracterizado (i.e. muchos eventos). Supongamos que la primer partida, juegan entre si los dos jugadores nuevos, donde **a** le gana a **b**. Al poseer mucha incertidumbre sobre estos jugadores, la actualización de habilidad de ambos es leve. En la siguiente partida, se enfrentan **b** y **c**, en donde **b** sale victorioso. Debido a que el jugador ganó un evento contra un jugador en el que se tiene certeza de que es muy bueno, la actualización de habilidad de **b** será grande. Sin importar el tiempo transcurrido entre estas dos partidas, intuitivamente podemos notar que del primer evento el jugador **a** ganó una partida más importante a la considerada por el modelo. Éste ganó una partida contra un jugador potencialmente hábil.

En ambos ejemplos, el algoritmo de **TrueSkill Through Time** soluciona estos problemas al permitir que la información se propague a ambos sentidos. Esto produce una actualización de las estimaciones de habilidad de todos los eventos de forma iterativa hasta su convergencia. Gracias a esto, el modelo logra estimar la misma habilidad a los tres jugadores del primer ejemplo y caracterizar a un agente con una menor cantidad de eventos.

Capítulo 2

Fundamentos

Este capítulo se encuentra dividido en dos partes. En la primer sección veremos los fundamentos necesarios para entender el método gráfico probabilístico de **TrueSkill**. Primero se introducirá un modelo probabilístico bayesiano que es usado como marco de referencia. Luego veremos algunas propiedades de las distribuciones gaussianas que resultan fundamentales para el tratamiento del método en cuestión. A su vez, veremos algunos conceptos como los gráficos de factores, pasaje de mensajes, aproximaciones utilizadas, formas de evaluación, entre otros. Se finaliza explicando el funcionamiento de **TrueSkill** y de su sucesor **TrueSkill Through Time**, que es el método utilizado en la parte experimental de esta tesis. Por último, se realiza un breve ejemplo detallado de las operaciones realizadas por **TrueSkill** que muestran los conceptos necesarios para lograr adquirir el manejo de estos métodos.

Si el lector contara con conocimientos de probabilidad y estadística, puede avanzar directamente a la sección 2.2 ([Modelos](#)) en la página 19. En particular, la sección 2.1.1 ([Modelo probabilístico](#)) y 2.1.2 ([Distribución gaussiana](#)) se consideran como conocimientos básicos.

En la segunda sección veremos algunos de los diversos tipos de métodos y modelos predictivos existentes para estimar e inferir variables ocultas. En lo que concierne a esta tesis, nos centraremos en los modelos de habilidad en juegos y deportes, y en su contextualización para interpretar el método **TrueSkill Through Time** y sus predecesores.

2.1 Introducción teórica

2.1.1 Modelo probabilístico

Uno de los primeros desafíos que se tiene es poder ser capaces de estimar variables de las cuales desconocemos su valor, lo cual nos lleva al concepto de incertidumbre. En el caso de **TrueSkill**, cuando estimamos la habilidad de un nuevo jugador, comenzamos con una creencia con mucha incertidumbre debido a que no tenemos información previa y, a medida que va participando en las diferentes partidas, vamos teniendo mayores elementos para inferir su habilidad y reducir así su incertidumbre. En general, es más probable que el jugador más *fuerte* sea el ganador, aunque ésto no está garantizado. El uso de sistemas de referencias probabilísticos nos permite cuantificar estas cantidades inciertas manejando de

manera sólida la incertidumbre asociada.

La probabilidad asociada a un suceso o evento aleatorio es una medida del grado de certidumbre de que dicho suceso ocurra. Los diferentes valores que se pueden dar en se los conoce como *variables aleatorias*. Utilizaremos la notación estándar donde la p minúscula es la densidad de probabilidad de una variable continua en tanto que la P mayúscula se utilizará para denotar la distribución de probabilidad de una variable discreta. Representaremos con x e y a dos variables aleatorias.

Existen diferentes distribuciones de probabilidades que son mencionadas y utilizadas en esta tesis. A la distribución en donde la variable aleatoria tiene solo dos valores, se la conoce como distribución de **Bernoulli** o bien como **función indicadora** (\mathbb{I}). En los casos en donde se tiene absoluta certeza de la variable de dos estados (i.e. probabilidad del 100 %), se lo conoce como **punto de masa**. En general, esto último se utiliza cuando se observa una variable, conociendo así su valor real. La probabilidad condicional, $P(x|y)$, es la probabilidad de que suceda un evento dado que tenemos información a partir de otra variable y . Se dice que dos variables son **independientes** si vale que $P(x|y) = P(x)$. En caso contrario, se las denomina como variables **dependientes**. El concepto de independencia es importante para este tipo de modelos, ya que cualquier variable no incluida explícitamente en nuestro modelo se supone que es independiente a todas las otras variables. Es importante observar que la suma de los posibles valores de la variable condicionada x se encuentra normalizada, pero no así la suma con los valores posibles de la variable y . Se conoce como **probabilidad conjunta**, a la probabilidad de que dos o más variables aleatorias tengan un determinado valor, $P(x, y)$. Realizar la suma de todas las variables excepto una (en este caso, x), se lo conoce como **marginalización** de la variable x , y da como resultado la probabilidad de esta variable.

$$P(x) = \sum_{y \in Y} P(x, y) \quad \text{o} \quad p(x) = \int p(x, y) dy \quad (\text{regla de la suma})$$

Esto significa que cualquier distribución marginal $p(x)$ puede ser obtenida integrando la distribución conjunta $p(x, y)$ sobre todos los posibles valores de y . En el caso discreto, se trata de manera similar pero realizando la suma sobre los valores de Y . Otra propiedad muy importante de las distribuciones conjuntas es la conocida como **regla del producto**, que la relaciona con la probabilidad condicional y marginal:

$$p(x, y) = p(x|y)p(y) \quad (\text{regla del producto})$$

Toda inferencia probabilística, no importa cuán compleja sea, puede ser resuelta a través de estas dos expresiones: la [regla de la suma](#) y la [regla del producto](#). Además, cualquier distribución conjunta puede ser expresada como el producto de distribuciones condicionales uni-dimensionales. Utilizando esta regla junto con la propiedad de simetría $p(x, y) = p(y, x)$, se obtiene inmediatamente el [teorema de Bayes](#).

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (\text{teorema de Bayes})$$

Se conoce como modelo bayesiano, a un modelo estadístico en donde se utilizan probabilidades para representar todas las incertidumbres dentro de nuestro modelo, tanto las incertezas de entrada como de salida. En este tipo de modelos, se utilizan los conceptos de distribuciones de probabilidad a priori y a posteriori. Es decir, poder usar teoría

probabilística para representar nuestro conocimiento de alguna variable aleatoria, conocida como **prior** y luego de obtener nueva información, poder hacer inferencia, y actualizar esta distribución a la cual se la denomina **posterior**. El uso inferencial del teorema de Bayes juega un rol central en las técnicas modernas de aprendizaje estadístico. Esta expresión nos permite actualizar nuestras creencias sobre ciertas hipótesis, dado un modelo y los datos. Para reducir la incertidumbre sobre la habilidad oculta utilizando la información provista por el resultado observado y por el modelo causal descripto, se requiere resolver la siguiente ecuación,

$$\underbrace{p(\overbrace{\text{Skills}}^{\text{Hidden}} \mid \overbrace{\text{Result, Model}}^{\text{Observed}})}_{\text{Posterior}} = \frac{\overbrace{P(\text{Result} \mid \text{Skills, Model})}^{\text{likelihood}} \overbrace{p(\text{Skills})}^{\text{Prior}}}{\underbrace{P(\text{Result} \mid \text{Model})}_{\text{Evidence}}} \quad (2.1)$$

donde el lugar que ocupa el modelo queda implícito. Se conoce como la función *likelihood* a la medida de la bondad del ajuste de nuestro modelo luego de haber inferido con una muestra de datos (resultado de un juego) para valores dados de las variables desconocidas (habilidad de los jugadores, *skill*). En nuestro caso, la hipótesis es el valor de la habilidad a priori de los jugadores en un determinado evento y el dato es el resultado de dicho juego (*result*). La evidencia es simplemente una constante de normalización, para que la probabilidad a posteriori lo esté.

2.1.2 Distribución gaussiana

El método de **TrueSkill** se puede utilizar para obtener estimaciones de la habilidad y otras variables con distribuciones gaussianas. En teoría de probabilidad, la distribución gaussiana, o también conocida como distribución normal, es un tipo de distribución de probabilidad continua para una variable aleatoria de valor real. Este tipo de distribución resulta muy útil debido a su simplicidad. Solamente es necesario utilizar dos parámetros para describirla: la media (μ) y la desviación estándar (σ). La primera representa el valor más probable de la variable aleatoria, y la desviación estándar representa la incerteza que se tiene sobre este valor. A su vez, estas distribuciones poseen diversas propiedades matemáticas que facilitan su manipulación algebraica, especialmente cuando se trabaja con múltiples distribuciones gaussianas. Su forma general está dada por,

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.2)$$

Donde la notación muestra la variable aleatoria x , seguido por los parámetros de la distribución μ y σ (análoga a la notación $x \sim \mathcal{N}(\mu, \sigma^2)$). El parámetro μ representa la media de la distribución, mientras que el parámetro σ la desviación estándar, la cual representa el ancho de la campana. La figura 2.1 presenta un esquema gráfico de una distribución normal o gaussiana típica definida en la ecuación 2.2.

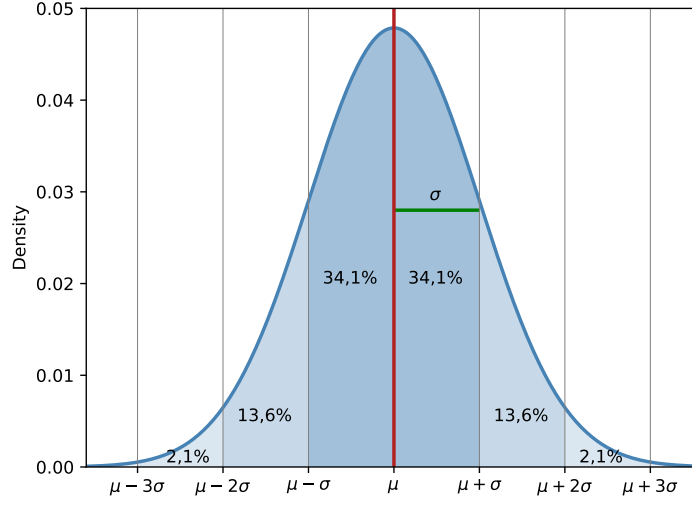


Figura 2.1: Distribución normal de media μ y desviación estándar σ . Los porcentajes representan el porcentaje de área correspondiente al sector sombreado.

Esta distribución está normalizada y además el área bajo la curva encerrada en una desviación estándar a cada lado de la media da el 68,2% del total. Por convención se define a σ^2 como la **varianza**. Hay varias propiedades que se utilizan para calcular **TrueSkill**, como son la suma y producto entre gaussianas, la función gaussiana acumulada, entre otras. Las demostraciones de estas propiedades se encuentran en el apéndice A. Estas distribuciones cuentan con la siguiente simetría:

$$\mathcal{N}(x|\mu, \sigma^2) = \mathcal{N}(\mu|x, \sigma^2) = \mathcal{N}(-\mu|-x, \sigma^2) = \mathcal{N}(-x|-\mu, \sigma^2) \quad (2.3)$$

La integral de la multiplicación de dos gaussianas está definida según la siguiente expresión:

$$\int_{-\infty}^{\infty} \mathcal{N}(x|\mu_x, \sigma_x^2) \mathcal{N}(x|\mu_y, \sigma_y^2) dx = \mathcal{N}(\mu_x|\mu_y, \sigma_x^2 + \sigma_y^2) \quad (2.4)$$

Se define como $\Phi(x|\mu, \sigma^2)$ a la función de acumulada gaussiana. Ésta tiene los mismos parámetros que la anterior distribución, integrándose desde $-\infty$ hasta x . En la figura 2.2 se puede visualizar una distribución gaussiana junto con la acumulada para todo x . Esta definida como,

$$\Phi(x|\mu, \sigma^2) = \int_{-\infty}^x \mathcal{N}(y|\mu, \sigma^2) dy \quad (2.5)$$

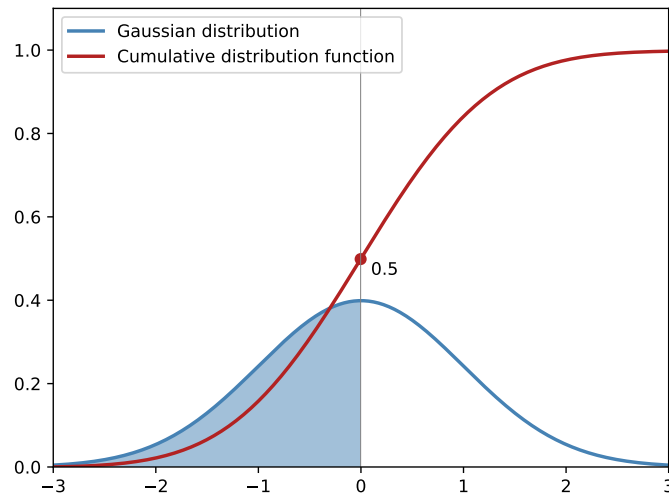


Figura 2.2: La curva azul muestra una distribución gaussiana con media cero y una desviación estándar de uno. El área bajo la curva de esta desde $-\infty$ hasta un punto x es conocido como distribución gaussiana acumulada y se muestra como una función de x por la curva roja. Se muestra el valor que tiene esta función en el valor 0, marcando el área integrada de la distribución gaussiana.

Podemos observar como el valor de la función acumulada representa el área bajo la curva de la distribución gaussiana desde $-\infty$ hasta el valor de evaluación. Al ser una distribución normalizada, el máximo valor es uno. En la figura 2.2 podemos observar el valor que se obtiene al evaluar la función acumulada en el valor de la media (cero en este ejemplo).

2.1.3 Gráfico de factores

A medida que incrementamos la cantidad de variables aleatorias en nuestro modelo, la visualización intuitiva tan sencilla como puede ser en el caso de solo dos variables, se vuelve más complicada. TrueSkill toma provecho de que la mayoría de las distribuciones conjuntas pueden escribirse como un número finito de términos o factores (i.e. un número pequeño de variables), pudiendo representar distribuciones conjuntas complejas usando gráficos de factores. Éstos muestran los distintos factores que forman la distribución junto con sus variables. Se utilizan dos tipos de nodos en estos gráficos. Por un lado se encuentran los nodos de variables, habiendo uno por cada variable del modelo y el nodo de factor, donde hay uno por cada factor en la distribución conjunta. Estos factores también se pueden interpretar como partes de la factorización de una función, pudiéndose aplicar a otros casos. En general se representan a los nodos de variables como círculos, encerrando el nombre de la variable y como cuadrados negros con una etiqueta del factor, a los nodos de factores. Se conecta cada factor con los nodos de variables en las cuales éste depende.

El siguiente ejemplo (figura 2.3) se tiene una función de tres variables, en la cual se puede representar como un único factor, o mediante la factorización de ésta, pudiendo quedar en tres factores. Se observa que cada factor está conectado con una línea con cada variable dependiente.

$$f(x, y, z) = (x + y)(y + z)(x + z)$$

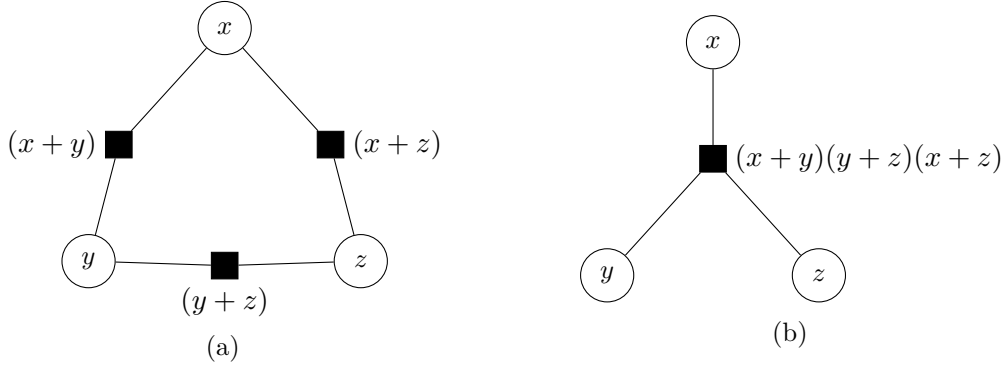


Figura 2.3: Se muestra dos formas distintas de representar la función $f(x, y, z) = (x+y)(y+z)(x+z)$ mediante gráficos de factores. Los círculos representan los nodos de variables y los rectángulos negros los factores. Cada factor se encuentra conectado con una línea con los nodos de las variables de las que depende explícitamente.

Tal como se muestra en el ejemplo, una función no tiene una única forma de factorización, y es por eso que existen diversos gráficos de factores para cada caso. La elección dependerá del grado de abstracción que se requiera.

Representar un modelo probabilístico usando un gráfico de factores, trae varios beneficios. Provee una manera simple de visualizar la estructura del modelo probabilístico y ver qué variables tienen relaciones entre ellas. Se puede usar para motivar y diseñar nuevos modelos al modificar cambios apropiados al gráfico. Las suposiciones codificadas dentro del modelo pueden ser vistas de manera clara. La información sobre las propiedades de un modelo se pueden obtener mediante operaciones realizadas de manera gráfica. Los cálculos realizados sobre el modelo pueden ser hechos de manera eficiente por un algoritmo que use la estructura del gráfico. En el caso de **TrueSkill**, el propio modelo es una gran probabilidad conjunta que representaremos gráficamente mediante este recurso.

2.1.4 Pasaje de mensajes

El concepto de pasaje de mensajes es parte fundamental para poder entender el método de **TrueSkill**. Esto se basa en que, a medida que nuestros gráficos de factores se vuelven más complejos, es necesario poder realizar el cómputo de inferencia de manera mecánica mediante algún algoritmo teniendo en cuenta, a su vez, el uso de recursos de cómputo. Esto se logra mediante el pasaje de información a través de los bordes del gráfico de factores, donde un **mensaje** es una distribución de probabilidad sobre las variables que están conectadas con ese borde.

En nuestro caso usaremos el algoritmo llamado **sum-product-algorithm** [20] más bien conocido en el área de estadística como **belief propagation** [21]. Fue desarrollado para computar distribuciones marginales posteriores sobre las variables en un gráfico de factores que no tengan ciclos o bucles. Este algoritmo usa dos tipos de mensajes distintos, uno para el mensaje que va de un factor a una variable, y otro para los mensajes que van de una variable a un factor. Las variables observadas envían un mensaje de punto de masa. Con esta herramienta, vamos a poder automatizar el cálculo de inferencia.

Sea $m_{x \rightarrow f}(x)$ el mensaje enviado por el nodo variable x al nodo factor f , y $m_{f \rightarrow x}(x)$

el mensaje enviado por un nodo factor f a un nodo variable x . Denotamos como $n(v)$ al conjunto de nodos vecinos al nodo v . Luego, el primer tipo de mensaje puede ser expresado del siguiente modo,

$$m_{x \rightarrow f}(x) = \prod_{h \in n(x) \setminus \{f\}} m_{h \rightarrow x}(x) \quad (2.6)$$

Es decir todos los mensajes que llegan al nodo x sin contar el que envía el nodo al cual se quiere enviar el mensaje. El segundo tipo de mensaje puede ser expresado como,

$$m_{f \rightarrow x}(x) = \int \cdots \int \left(f(\mathbf{x}) \prod_{h \in n(f) \setminus \{x\}} m_{h \rightarrow f}(h) \right) d\mathbf{x}_{\setminus x} \quad (2.7)$$

donde $\mathbf{x} = \arg(f)$ es el conjunto de argumentos de la función f . Este mensaje es parecido al anterior con la diferencia que se multiplica por su factor y se integra por todas las variables exceptuando a la que se está enviando. Luego, para calcular una marginal cualquiera, se multiplican todos los mensajes que llegan a dicho nodo:

$$p(x_i) = \prod_{h \in n(x_i)} m_{h \rightarrow x_i} \quad (2.8)$$

El uso de esta última es de suma importancia para el método. Es la marginal la cual estima la habilidad de los jugadores, mediante la variable de dicho parámetro. Por otro lado, una propiedad en la cual usaremos seguido en los mensajes, son las integrales con una función indicadora. En **TrueSkill** se utiliza dicha función para restringir el espacio de probabilidades al observar el resultado de un evento y conocer el ganador.

Integrales con función indicadora

$$\begin{aligned} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbb{I}(x = h(y, z)) f(x) g(y) dx dy &= \int_{-\infty}^{\infty} \int_{h(y, z)}^{h(y, z)} f(h(y, z)) g(y) dx dy \\ &= \int_{-\infty}^{\infty} f(h(y, z)) g(y) dy \end{aligned} \quad (2.9)$$

Podemos ver que gracias a esta función, nos queda una integral de una única variable. En el caso de esta tesis, se utilizan los gráficos de factores junto con el pasaje de mensajes. Veamos un ejemplo en que se requiere estimar la media m de una distribución gaussiana con desviación estándar uno, mediante observaciones independientes x , usando inferencia bayesiana, representándolo con un gráfico de factores como se observa en figura 2.4. Esta estimación se puede representar algebraicamente utilizando la [regla del producto](#) en donde la distribución conjunta es igual al *prior* del parámetro m , multiplicado por la productoria de los *likelihoods* de cada observación (probabilidad condicional de x_i dado m). Estamos suponiendo que la distribución condicional de x_i es una gaussiana con media m y varianza uno.

$$\begin{aligned} p(m, x_1, \dots, x_n) &= p(m) \prod_i p(x_i | m) \\ p(x_i | m) &= \mathcal{N}(x_i | m, 1) \end{aligned}$$

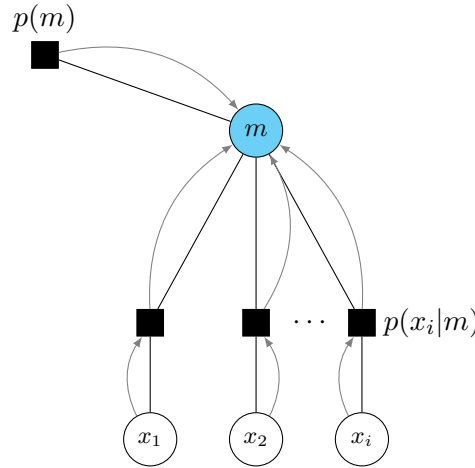


Figura 2.4: Representación grafica por mediante a factores de la estimación de la media de una distribución gaussiana. Se sombrea en azul la variable a la cual se está calculando su marginal. Las flechas grises representan los mensajes enviados entre los nodos para el cálculo de la marginal en cuestión.

Podemos observar un factor, $p(m)$, que representa el *prior* de la variable m , otro nodo para dicha variable, y luego un factor por cada *likelihood*. El *prior* tiene que representar el conocimiento a priori de la hipótesis a inferir. Si suponemos que no tenemos información podemos elegir una distribución gaussiana con media cero y una desviación estándar grande, como puede ser $\mathcal{N}(m|0, 100)$. Cada *likelihood* representa la verosimilitud de obtener el valor observado x_i suponiendo que la hipótesis m es verdadera. Dicho de otra forma, el *likelihood* evalúa la probabilidad de la observación x_i para las diferentes hipótesis m . En los casos en los que la hipótesis ofrece muy baja probabilidad del valor observado (cuando se encuentra muy alejada de nuestra observación) el *likelihood* será cercano a 0 y actuará como filtro reduciendo su densidad, produciendo el efecto contrario mientras mas cercanos se encuentren los valores. Al tomar la marginal del nodo m (sombreado en azul), obtendremos el *posterior* de dicha variable. Es decir, actualizaremos nuestra creencia de la media de nuestra distribución luego de obtener las observaciones.

Gracias al modelo gráfico, debido a que cada factor solo se conectan con la variable m y no entre sí, se desprende visualmente que no hay dependencia entre ellos. Otra de las ventajas de este modelo gráfico es que logra representar de forma gráfica el funcionamiento del algoritmo. En este ejemplo, cuando inferimos sobre m , vemos que los nodos x_i deben enviar mensajes al nodo m . Aun con gráficos de factores complejos, la probabilidad conjunta sobre las variables aleatorias (representada por los nodos), puede ser escrita como el producto de los factores (representados por los nodos de factores). La probabilidad conjunta da una completa especificación del modelo, ya que define la probabilidad de cualquier posible combinación de valores para todos las variables aleatorias incluidas. En el caso de tener el valor de una variable aleatoria, éste se representa con el nodo de dicha variable sombreada en rojo, para enfatizar que es una variable observada.

2.1.5 Expectation Propagation

Uno de los inconvenientes con **belief propagation** es cuando existe un ciclo en el gráfico de factores. Un ciclo se puede presentar tanto en **TrueSkill** con eventos en los que participan tres o más equipos y también en **TTT**. Por como están formados, se requiere iterar

una sección del gráfico de factores en forma cíclica hasta la convergencia de los mensajes, es decir, hasta que la variación de los mensajes sea lo suficientemente chica. El valor límite (*threshold*) se lo define con anterioridad. Mientras menor sea este valor más iteraciones se requerirán y, por ende, aumentará el costo computacional. Para poder manejar un ciclo se requiere un orden sobre cómo y cuándo los mensajes se envían. En particular, en las partes en donde se iteran los mensajes hasta su convergencia. En el caso de **TrueSkill**, se utiliza el método de **Expectation Propagation** [22], introducido por Tom Minka. Éste permite aproximar unos de los mensajes de modo tal que la forma funcional del *posterior* sea igual que el del *prior*. La ventaja de preservar la forma funcional de estas distribuciones es que, debido a que el *posterior* de un jugador en **TrueSkill** será el *prior* de la partida siguiente de dicho jugador, se logra conservar la cantidad de parámetros de nuestro modelo para describir la habilidad. En caso contrario, a medida que el algoritmo evolucione con más cantidad de partidas, la forma funcional iría variando, requiriendo cada vez un mayor número de parámetros para poder describir la habilidad, impactando fuertemente en la mecánica de resolución. La aproximación se verá más adelante en la sección de **TrueSkill** en mensaje aproximado.

2.1.6 Metodología de evaluación

Un aspecto importante es poder tener alguna forma de medir y comparar los modelos de forma numérica. Para ello se debe elegir una **medida de evaluación** que sirva para comparar las estimaciones obtenidas. Una forma común de hacer esto es evaluar la probabilidad de los valores que fueron observados en la realidad bajo las distribuciones inferidas. En el caso de esta tesis, corresponde con la probabilidad de que el resultado de una partida ocurra considerando las habilidades a priori estimadas de los jugadores involucrados. Si la probabilidad del equipo que ganó fuera baja según nuestras estimaciones, indicaría que las distribuciones a priori no captan la esencia de la realidad. En caso contrario, al coincidir nuestra estimación con el resultado que ocurrió, aumentaría la validez de nuestras creencias.

Siguiendo este razonamiento, podemos obtener un mecanismo para comparar cuantitativamente las inferencias sobre un evento a partir de distintos modelos. En base al resultado observable y a las habilidades a priori obtenidas por cada modelo, lo más habitual es que cada uno de los modelos presente una probabilidad distinta de que el jugador/equipo ganador salga victorioso. El modelo con mayor valor de probabilidad será el que consideramos como el de la mejor predicción. De esta manera, más allá de que ambos modelos pudieran predecir el resultado de la partida, obtenemos una manera cuantitativa para comparar ambos modelos. Además, en los casos en que la probabilidad sea muy baja buscaremos penalizar al modelo fuertemente.

En el caso de contar con múltiples partidas, podemos realizar un seguimiento de estas probabilidades en toda la base de datos mediante su multiplicación. A esta productoria la denominamos como **evidencia**. Debido a que son probabilidades, el valor máximo de ésta es uno, representando una eficiencia del 100 % de nuestro modelo. El mínimo valor es cero, representando una nula concordancia entre nuestro modelo y los datos. Al tener valores entre cero y uno, se aplica el logaritmo a toda la productoria, generando números más manejables cuando la evidencia es muy baja. Además, por propiedades del logaritmo, la productoria se transforma en una suma, lo cual es computacionalmente más eficiente. La peor predicción nos dará una probabilidad de cero, es decir $\log(0) = -\infty$ y la perfecta predicción nos dará una probabilidad de uno dando $\log(1) = 0$. Como ningún sistema real es perfecto, en la práctica el logaritmo de la probabilidad siempre será un valor negativo. Por conveniencia, cambiamos el signo de la sumatoria de logaritmos, obteniendo

así valores positivos y, mientras menor sea el valor total, mejor es la predicción del modelo. De esta forma, definimos a la evidencia como menos la sumatoria de los logaritmos de las probabilidades de ganar de los jugadores ganadores (P_i), dividido por la cantidad de partidas (N), para tener un promedio y un número más manejable.

$$\text{evidencia} = -\frac{\log(\prod_{i=1}^N P_i)}{N} \quad (2.10)$$

En general esta medida penaliza fuertemente las predicciones erróneas, ya que el logaritmo da valores negativos muy grandes cuando la probabilidad de los valores reales es muy cercana a cero. En particular, ésto tiene que ser tomado en cuenta cuando tenemos una base de datos en las cuales sus valores contienen un cierto error. En el caso como el de esta tesis, donde se toman datos reales (si el juego lo ganó un jugador/equipo u otro), se considera que no hay error en los datos, haciendo esta medida muy eficiente. En la siguiente sección, luego de mostrar las bases del método, podremos ver cómo calcular estas probabilidades y así obtener la evidencia.

■ 2.2 Modelos

■ 2.2.1 Elo

El modelo de ranking de Elo fue uno de los primeros en crearse. Fue desarrollado por Arpad Elo en 1959 [8,10]. La idea central de Elo es modelar la habilidad de los jugadores por medio con un único valor denominado s_i y luego utilizarlo para la creación de un ranking y, eventualmente, para predecir resultados de partidos. En particular, maneja el empate como la mitad de una victoria y la mitad de una derrota, sin predecir la probabilidad de que ocurra este hecho. Este método fue adoptado en 1970 por la Federación mundial de ajedrez (FIDE).

Sin embargo, la habilidad de un jugador es una cantidad que tiene incertidumbre asociada. Una de las suposiciones de Elo es tratar a la habilidad como un único valor que se actualiza cuando se observa el resultado de un juego. Dado un resultado de una partida, entendemos que sería razonable aumentar el valor de habilidad del ganador y disminuir el valor de la habilidad del perdedor. Lo que no queda claro es cuánto hay que ajustar estos valores. Intuitivamente podemos pensar lo siguiente: si el jugador que ganó posee una habilidad mucho mayor que el de su contrincante, entonces no nos sorprende este resultado, y el cambio de habilidades tendría que ser relativamente pequeño. En cambio, si la habilidad del jugador ganador fuera significativamente más chica que su rival, entonces el resultado de que éste gane sorprendería mucho, habilitando una actualización grande de las habilidades de ambos jugadores. Es importante entender como el grado de **sorpresa** nos indica qué tan grande debe ser el cambio en la actualización de los valores.

El modelo supone que cada jugador exhibe un rendimiento al momento de cada evento. El jugador que exhibe mayor rendimiento es quien resulta ganador. El rendimiento engloba todos los factores que afectan al jugador y se lo representa como una variable aleatoria normalmente distribuida centrada en el verdadero valor de habilidad junto con una constante de ruido β : $p_i \sim \mathcal{N}(s_i, \beta^2)$. Este valor de ruido (β) se utiliza para generar la escala relativa del modelo. Por como esta formado el método, cuando dos jugadores presentan una diferencia de habilidad de un β , el jugador más fuerte tiene un 76 % de

probabilidad de ganar. Bajo estas suposiciones, podemos inferir en cada juego quién tuvo el rendimiento más grande, al observar el resultado real de la partida, (gano/perdió). Luego, la probabilidad de que el jugador i gane es $p(p_i > p_j | s_i, s_j) = p(d_{ij} = p_i - p_j > 0 | s_i, s_j)$. La probabilidad de una cierta diferencia de rendimiento d_{ij} se computa como:

$$p(d_{ij} | s_i, s_j) = \iint \mathbb{I}(d_{ij} = p_i - p_j) \mathcal{N}(p_i | s_i, \beta^2) \mathcal{N}(p_j | s_j, \beta^2) dp_i dp_j \quad (2.11)$$

En la figura 2.5 se muestra el espacio de rendimientos en el caso de dos jugadores, representado mediante el integrando de la ecuación 2.11. El valor de β condiciona el ancho de dicha distribución.

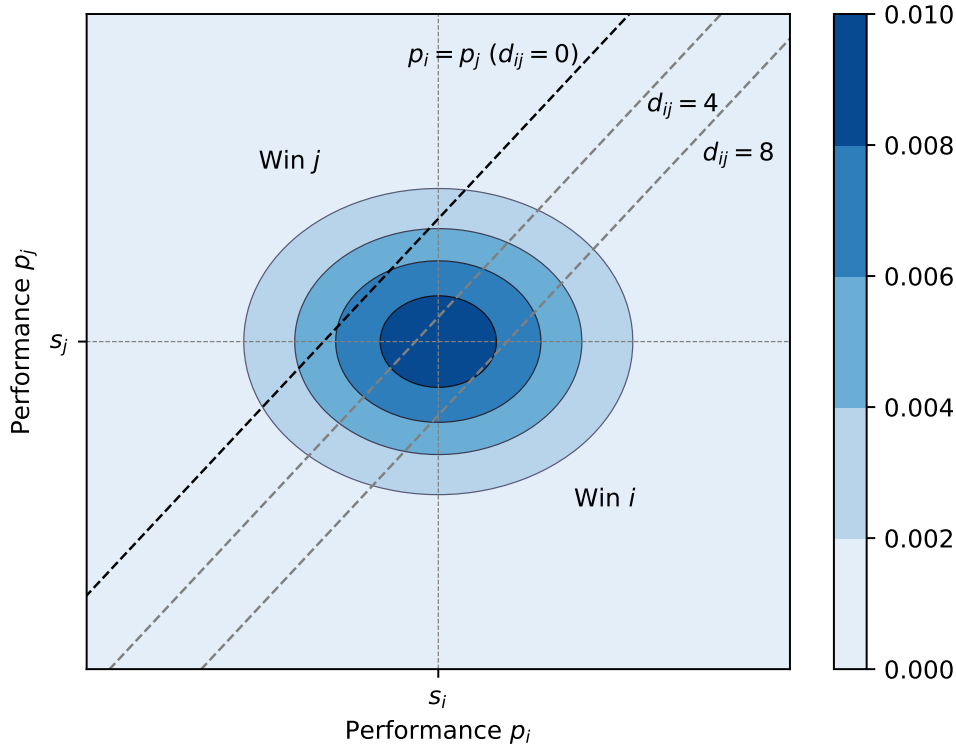


Figura 2.5: Probabilidad conjunta del rendimiento (proyección de una gaussiana en tres dimensiones) de dos jugadores, i, j suponiendo que $s_i > s_j$. La línea diagonal $p_i = p_j$ separa el espacio de probabilidades entre las dos posibilidades de victoria. Considerando el caso donde el jugador i es el ganador, se muestra sombreado con rojo el caso contrario. Todas las líneas paralelas a la diagonal $p_i = p_j$ representan las líneas de contorno de diferencia de rendimiento $d_{ij} = p_i - p_j$ constante.

Se muestra que el espacio de probabilidades es una distribución gaussiana de dos dimensiones. La densidad de la misma se encuentra representada mediante un mapa de color. Las líneas de contorno de diferencia de rendimiento $d_{ij} = p_i - p_j$, son líneas paralelas a la diagonal $p_i = p_j$. Basándonos en las propiedades de las gaussianas, se puede mostrar que la diferencia de rendimiento d_{ij} , también está distribuida normalmente, centrada en la diferencia de la habilidad, con el doble de varianza, como se observa en la figura 2.6.

$$p(d_{ij}|s_i, s_j) = \mathcal{N}(d_{ij}|s_i - s_j, 2\beta^2) \quad (2.12)$$

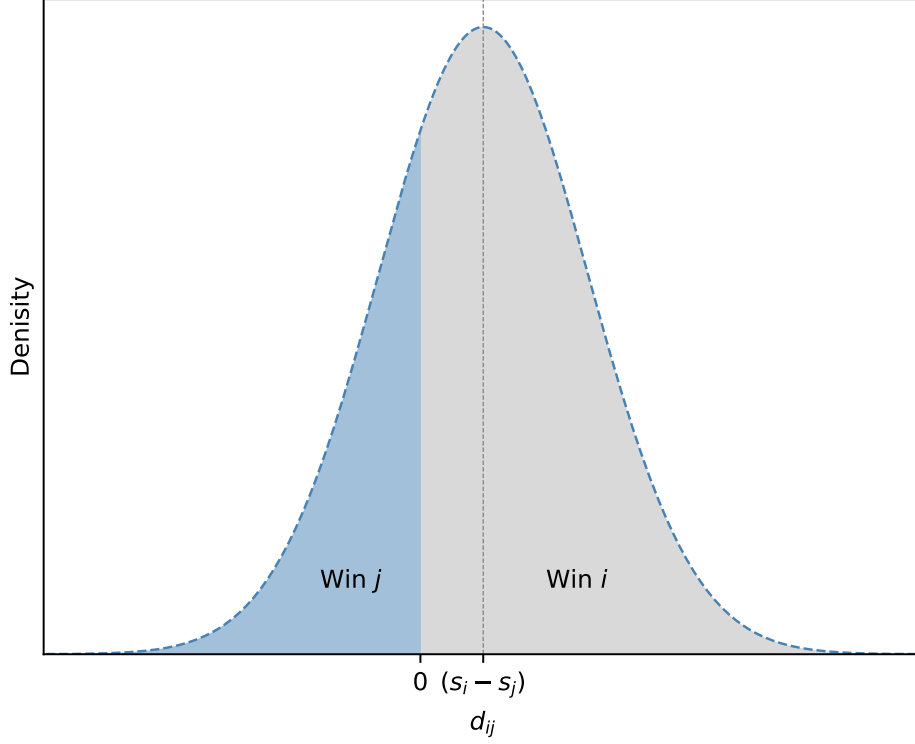


Figura 2.6: La probabilidad del resultado de un juego bajo la suposición del sistema de Ranking Elo con $s_i > s_j$. El área bajo la curva en el intervalo positivo ($d_{ij} > 0$) es la probabilidad de que el jugador i gane y el área bajo la curva en el intervalo negativo es la probabilidad de que el jugador j gane.

Esto reduce el problema de computar la probabilidad del resultado del juego a un problema unidimensional relacionado a la diferencia de rendimiento. Sea el resultado del juego r_{ij} . Se puede computar la probabilidad de que el jugador i sea el ganador, $r_{ij} = 0$ ($r_{ij} = 1$ en caso contrario), como,

$$P(r_{ij} = 0|s_i, s_j) = p(d_{ij} > 0|s_i, s_j) = 1 - \Phi\left(0|s_i - s_j, 2\beta^2\right) \quad (2.13)$$

donde Φ es la función de distribución acumulada hasta cero de la distribución normal estándar, $\mathcal{N}(x|s_i - s_j, 2\beta^2)$. La igualdad destacada (*), se deriva de la simetría que tienen las función de densidad normal. Luego, la probabilidad del resultado se puede escribir de la siguiente manera,

$$P(r_{ij}|s_i, s_j) = r_{ij}P(r_{ij} = 1|s_i, s_j) + (1 - r_{ij})P(r_{ij} = 0|s_i, s_j) \quad (2.14)$$

Con esto podemos calcular la probabilidad de un resultado dadas las estimaciones de la habilidad (s_i, s_j) . La solución metodológica es extremadamente simple y astuta: se empieza con una estimación arbitraria y luego de ver nueva información se actualizan en base a la

sorpresa generada por el resultado (i.e. un jugador débil le gana al mejor). La predicción a priori inducida por la estimación previa s_i y s_j , definen la sorpresa generada,

$$\Delta_i = \underbrace{y_{ij}}_{\substack{\text{Direction} \\ \text{(Outcome)}}} \underbrace{(1 - P(r_{ij}|s_i, s_j))}_{\substack{\text{Magnitud} \\ \text{(Outcome Surprise)}}} \quad (2.15)$$

Donde Δ_i es la sorpresa generada, y_{ij} es 1 si el agente i es el ganador y -1 en caso contrario, definido como $y_{ij} = 2r_{ij} - 1$. La magnitud de la sorpresa está relacionada con la probabilidad de que se obtenga el resultado. Si el resultado del evento es el esperado, entonces la sorpresa generada va a ser baja. En caso contrario, cuando nos encontramos con un resultado poco esperado, entonces la sorpresa será grande. Esta sorpresa es usada para actualizar la estimación de habilidad que se tenía de cada jugador hasta entonces:

$$s_i^{\text{new}} = s_i^{\text{old}} + \Delta_i \quad (2.16)$$

donde s_i^{new} es la habilidad actualizada del jugador i . De la misma forma, s_i^{old} es la estimación previa y Δ_i es la sorpresa generada. De forma análoga se realiza la actualización para el agente j utilizando la sorpresa Δ_j con signo opuesto. Este procedimiento logra recuperar la escala relativa de los agentes, empezando desde un valor inicial arbitrario. Sin embargo, esta metodología presenta un gran inconveniente operacional. La regla de actualización en la ecuación 2.16 es simétrica: lo que un agente pierde el oponente lo obtiene. Al no poder tener estimaciones precisas de un jugador nuevo (se usa un valor arbitrario fijo para todos los jugadores nuevos), el primer juego tiende a generar mucha sorpresa. Estos juegos pueden modificar abruptamente habilidades que ya habían sido estimadas de forma precisa. Para romper esta simetría de la ecuación 2.16, se propuso una solución ad-hoc: reducir el impacto de la sorpresa Δ_i basado en el número de eventos que en los que participó el agente previamente. Este es el rol que juega el factor K , usado en FIDE, $s_i^{\text{new}} = s_i^{\text{old}} + K_i \Delta_i$.

Además de las limitaciones de tener un parámetro arbitrario K , el cual tiene que ser definido previamente, este método cuenta con un segundo problema, en que la habilidad estimada tiene que ser considerada *provisional* hasta que el jugador alcance una cantidad arbitraria de participaciones. Un tercer problema es que con Elo no se puede estimar la habilidad de los jugadores cuando se juega en equipo ni tampoco se puede estimar la probabilidad de un empate. Varias modificaciones al sistema Elo básico fueron propuestos, en su mayoría mediante técnicas ad-hoc, para poder manejar algunas de sus limitaciones. Como ya se comentó, un empate puede ser representado como una puntuación 1/2 a la hora de computar la actualización de habilidad. El modelo bayesiano de TrueSkill resuelve todos estos problemas.

2.2.2 TrueSkill

El método de TrueSkill fue introducido en el 2006 por Ralf Herbrich [9]. Éste comparte el modelo de dependencia del sistema de calificación Elo entre habilidad, rendimiento y probabilidad de ganar. Es extendido mediante un modelo bayesiano que incorpora una distribución a priori de habilidad (*prior*), un modelo de rendimiento de equipos y una función de actualización no arbitraria (*posterior*). Debido a que la habilidad es una cantidad incierta, es incluida al modelo como una variable aleatoria. La distribución *prior* apta para esta variable, es la distribución normal, la cual captura el conocimiento a priori sobre el jugador antes de que juegue una partida. Ésta logra también capturar la noción de

incertidumbre en la habilidad debido a su parámetro de desviación estándar. Como vemos a diferencia de Elo, una de las suposiciones de este nuevo modelo es que cada jugador tiene un valor de habilidad representado por una variable continua con una distribución gaussiana y no por un escalar.

Una vez que un jugador haya disputado una partida, se utiliza el resultado de dicho evento para inferir y actualizar las distribuciones de habilidad de los jugadores involucrados. Ésto implica resolver un problema de inferencia probabilística para calcular la distribución *posterior* de cada jugador, tomando en cuenta la nueva información proveída por el resultado de la partida. A pesar de que nuestra distribución *prior* es gaussiana, la correspondiente *posterior* puede no serla. Este es el caso de usar el gráfico de factores de **TrueSkill**, e inferir de manera exacta. El beneficio de aproximar este *posterior* mediante una gaussiana se debe a que esta distribución de un jugador va a actuar efectivamente como la distribución *prior* de la próxima partida. Es por ello que interesa preservar su forma funcional. En caso contrario, por cada nueva estimación la forma funcional que describe la habilidad cambiaría, no solo complejizando las cuentas (por no poder operar con las propiedades algebraicas de las gaussianas), si no que a su vez aumentaría el número de parámetros para describir cada nueva distribución en cada nuevo evento. Una generalización del método **TrueSkill** se observa en la figura 2.7.

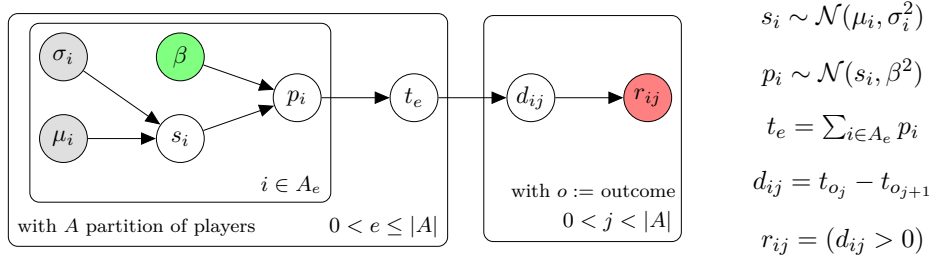


Figura 2.7: Representación esquemática del modelo generativo **TrueSkill**. Las habilidades ocultas causan los resultados observables a través de la diferencia de rendimientos ocultos. En cada evento, gana el agente quien haya obtenido mayor rendimiento, $r = p_i > p_j$. Esta representación gráfica define una distribución de probabilidad conjunta. Las variables observables se pintan de rojo, las ocultas en blanco, los parámetros constantes en verde y los parámetros de la distribución *prior* en gris.

Para el caso de un evento en donde se enfrentan dos jugadores, el grafo exacto se puede ver en la figura 2.8.

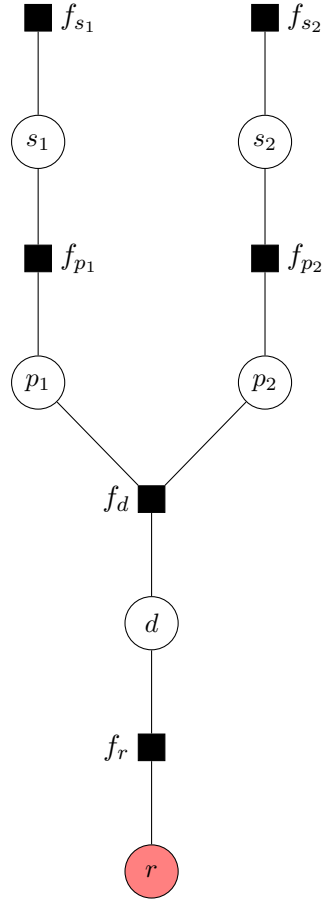


Figura 2.8: Factorización gráfica del modelo generativo (Fig. 2.7), para el caso particular de dos jugadores enfrentándose.

Podemos notar que, en este caso, no se requiere el uso de los nodos de equipo. Es importante entender, cómo funcionan estos grafos, para poder entender la intuición de éstos. Podemos observar gráficamente a todas las variables aleatorias de nuestro interés, tanto la habilidad, rendimiento como el resultado y un nodo de factor para cada una. Como ya se mostró en la sección introducción, existen dos tipos de mensajes. Los que van de nodos de variables a nodos de factores, y los contrarios. Utilizando la ecuación 2.8, se calculan los *posteriors* como las marginales de las variables de habilidad de los jugadores. Una de las consideraciones ya mencionadas, será el uso de *expectation propagation*, para aproximar los mensajes que hacen cambiar la forma funcional gaussiana a la marginal de habilidad. En la figura 2.9, se puede observar el flujo de mensajes para el cálculo de las marginales de la habilidad de ambos jugadores.

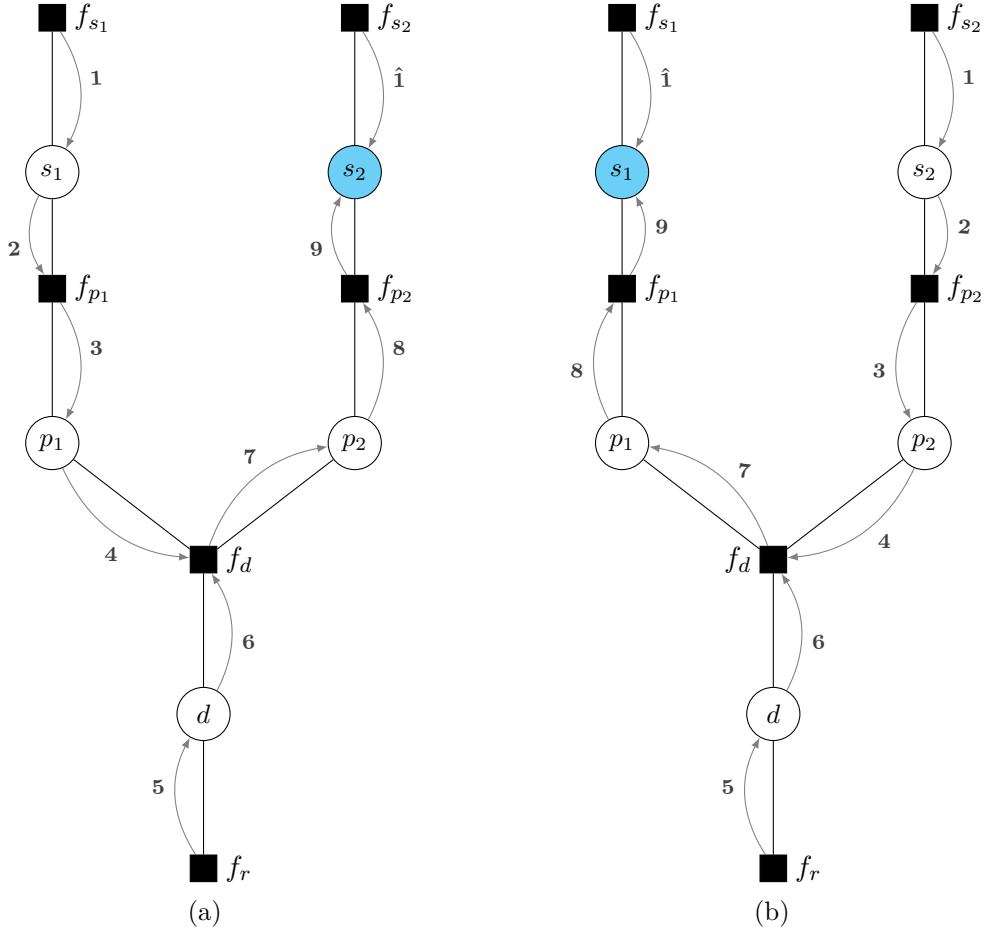


Figura 2.9: Factorización gráfica del modelo generativo para el caso particular de dos jugadores enfrentándose (Fig. 2.8). Se muestra el pasaje de mensajes para el cálculo de marginales para las variables de habilidad marcadas en azul. En (a) se muestra el cálculo de la *posterior* del jugador 2 y en (b) se muestra el cálculo de la *posterior* del jugador 1. Por cuestión de espacio, se omitió el nodo de la variable observada r .

Se puede observar el orden de los mensajes para poder calcular la *posterior* para el jugador 1 en la figura 2.9(a) y para el jugador 2 en la figura 2.9(b). Cabe recordar, por como esta formado el método, los jugadores/equipos son organizados por orden de ganador a perdedor. En este caso, implica que el jugador 1 fue el ganador del evento. Un ejemplo del cálculo de los mensajes y de cómo se calcula una marginal de habilidad se muestra en el apéndice B.2 (Ejemplo 1 vs 1-Handicap).

2.2.3 TrueSkill insights

A continuación veremos los diferentes conceptos y definiciones usadas en TrueSkill. El comportamiento de los diferentes parámetros de este método y las relaciones entre éstos se verá en la sección Validación en la página 41.

Habilidad Una de las novedades de TrueSkill es la noción de incertidumbre en la estimación de habilidad. La habilidad estimada del jugador i , s_i , que en el modelo anterior de Elo se representaba con un escalar, ahora se representa con una distribución *prior* de creencia con una función de distribución normal.

$$s_i \sim \mathcal{N}(\mu_i, \sigma_i^2) = \mathcal{N}(s_i | \mu_i, \sigma_i^2) \quad (2.17)$$

donde μ y σ , se inicializan con un valor arbitrario para todos los jugadores en su primer partida. El valor de inicialización del método lo denotaremos como μ_0 y σ_0 , siendo definidos previamente. Luego, por cada nuevo evento de cada jugador, estos valores se irán actualizando. Cabe resaltar que el valor absoluto de la media no es de interés en sí mismo, si no la diferencia con los demás jugadores. Por otro lado, la desviación estándar inicial tiene que ser lo suficientemente grande para poder representar la incertidumbre que realmente tenemos con respecto a la media, pero lo suficientemente pequeña para que se requiera una baja cantidad de eventos para caracterizar a un nuevo jugador. En esta tesis utilizaremos los valores iniciales, $\mu_0 = 25$ y $\sigma_0 = 3\beta$, donde el β vale uno en los primeros análisis. En la sección validación, veremos cómo la media inicial es un parámetro arbitrario el cual no modifica la escala relativa de las habilidades. El valor elegido de μ_0 como el valor de β uno, coinciden con los utilizados en el trabajo en el cual se presentó el método *TrueSkill*. La elección de la desviación estándar de la habilidad (σ) y del rendimiento (β) se estudia en la misma sección, eligiendo la mejor relación según la metodología de evaluación utilizada.

Rendimiento Al igual que el sistema Elo, se supone que el resultado final depende del rendimiento de los jugadores, p_i . Nuevamente, se representa como una distribución gaussiana con media igual a la habilidad y con una desviación estándar β .

$$p_i \sim \mathcal{N}(s_i, \beta^2) = \mathcal{N}(p_i | s_i, \beta^2) \quad (2.18)$$

La gran diferencia es que, ahora, el rendimiento depende de una habilidad modelada como una distribución gaussiana y no con un escalar. Por como está conformado el método, cuando la diferencia de habilidad entre dos jugadores es igual a β y la incertidumbre de éstos es nula (i.e. $\sigma=0$), la probabilidad que tiene el jugador de mayor habilidad en ganar es de un 76 %. Este parámetro es el que nos indica la escala relativa entre los jugadores, como se verá en la sección [Validación](#). La probabilidad que tiene un dado rendimiento p_i , que considera a la distribución *prior*, se define como:

$$p(p_i | \mu_i, \sigma_i) = \int \mathcal{N}(p_i | s_i, \beta^2) \mathcal{N}(s_i | \mu_i, \sigma_i^2) ds_i \quad (2.19)$$

Este resultado sale del pasaje de mensajes del gráfico de factores del modelo de la figura 2.8. Tanto la media del jugador i , μ_i , como su incerteza, σ_i , son escalares, mientras que p_i y s_i son variables aleatorias. Por cada valor de s_i hay una distribución gaussiana con dicha media, $\mathcal{N}(p_i | s_i, \beta^2)$. Esta distribución cumple el rol de peso para la distribución normal $\mathcal{N}(s_i | \mu_i, \sigma_i^2)$. Se representa gráficamente el integrando de este resultado en la figura 2.10.

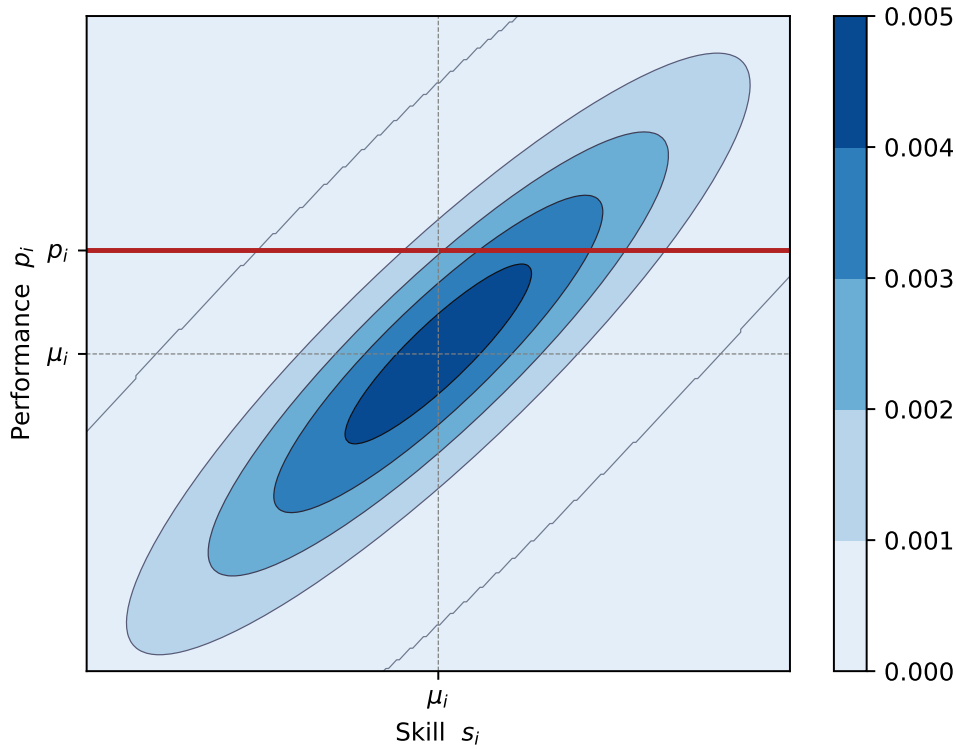


Figura 2.10: Distribución de habilidad, $\mathcal{N}(s_i|\mu_i, \sigma_i^2)$, pesado por la probabilidad de distribución de rendimiento $\mathcal{N}(p_i|s_i, \beta^2)$. El área bajo la curva sólida tiene que ser integrada para computar una cierta probabilidad p_i .

Se muestra nuevamente que el espacio de probabilidades es una distribución gaussiana de dos dimensiones. La densidad de la misma se encuentra representada mediante un mapa de color. Utilizando la propiedad de simetría, $N(x|\mu, \sigma^2) = N(\mu|x, \sigma^2)$, se reescribe el integrando de la ecuación 2.19, lo cual facilita su manipulación algebraica,

$$p(p_i|\mu_i, \sigma_i) = \int \mathcal{N}(s_i|p_i, \beta^2) \mathcal{N}(s_i|\mu_i, \sigma_i^2) ds_i \quad (2.20)$$

Luego, se computa la probabilidad de un dado rendimiento p_i integrando el área bajo la línea sólida de la figura 2.10. Utilizando la propiedad del producto de gaussianas correspondiente a la expresión 2.4, esta probabilidad también es una distribución normal.

$$p(p_i|\mu_i, \sigma_i) = \int \underbrace{\mathcal{N}(p_i|\mu_i, \beta^2 + \sigma_i^2)}_{\text{Scalar independent of } s_i} \mathcal{N}(s_i|\mu_*, \sigma_*^2) ds_i = \mathcal{N}(p_i|\mu_i, \beta^2 + \sigma_i^2) \quad (2.21)$$

Podemos observar que la probabilidad de un dado rendimiento p_i resulta en una distribución gaussiana similar a la distribución de habilidad. En este caso, la varianza es la suma de la varianza de la habilidad σ_i^2 y del rendimiento β^2 .

Equipos La segunda novedad de TrueSkill es la de poder estimar la habilidad de los jugadores cuando éstos juegan en equipos. El modelo de TrueSkill establece la suposición

de que un equipo le gana a otro cuando el rendimiento combinado del equipo es superior al rendimiento del equipo contrario. Para ello, al ejemplo mostrado en la figura 2.8, se agrega un nodo de factor de equipo junto con un nodo de la variable de rendimiento del equipo. Éstas conectan con el rendimiento de los miembros de dicho equipo al nodo diferencia, como se mostró en la figura 2.7. Otra suposición que realiza el método es que el rendimiento de un equipo se define como la suma de los rendimientos individuales de sus miembros. Sea A_e la partición de jugadores del equipo e (la asignación de equipo), queda definida entonces la variable de rendimiento de equipo t_e como:

$$t_e = \sum_{j \in A_e} p_j \quad (2.22)$$

Luego, mediante los mensajes, la probabilidad de un dado rendimiento de equipo se define como:

$$p(t_e | A_e) = \int \cdots \int \mathbb{I}(t_e = \sum_{j \in A_e} p_j) \left(\prod_{i \in A_e} \mathcal{N}(p_i | \mu_i, \beta^2 + \sigma^2) \right) d\vec{p} \quad (2.23)$$

La suposición acerca de cómo se modela el rendimiento de equipo es solo usada para adoptar la habilidad individual de jugadores de forma tal de que el resultado del equipo pueda ser mejor estimado en base a la suposición de adición de las habilidades. Esta suposición sirve para los juegos en los cuales el equipo ganador depende del conjunto de sus jugadores y no, como podría ser en las carreras de auto, que gana el equipo con el mejor rendimiento individual (aunque, como en el caso de las carreras de autos, hubiera un aspecto de competencia por equipos). A modo de ejemplo, en este último caso se debería cambiar la suposición de victoria, por una que diga que el equipo con mayor rendimiento individual sea el ganador. Matemáticamente, el rendimiento de un equipo con dos jugadores se puede observar gráficamente en la figura 2.11.

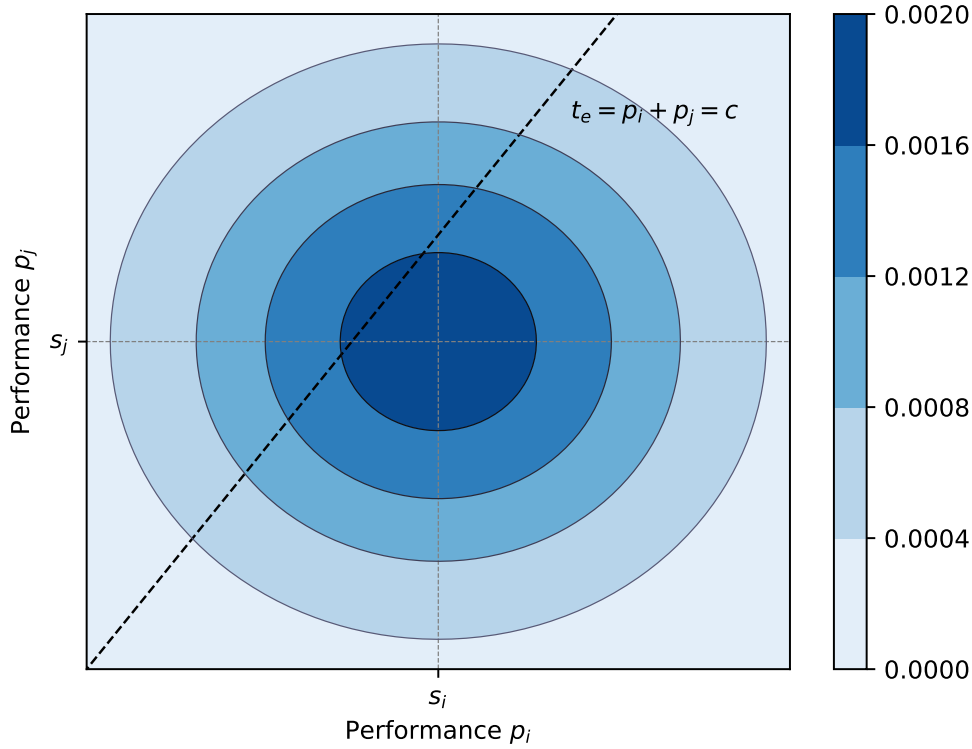


Figura 2.11: Probabilidad conjunta del rendimiento de dos integrantes i, j de un equipo. Las líneas paralelas a $t_e = c$ representan líneas de contorno de rendimiento de equipo.

Para computar la probabilidad de un dado rendimiento c de equipo, se tiene que integrar el área bajo la línea de contorno correspondiente, $t_e = c$ como se muestra en la figura 2.11. Obteniendo así una nueva gaussiana. En el caso de un equipo con dos jugadores, éste queda como:

$$\begin{aligned}
 p(t_e | A_e = \{i, j\}) &= \int \int I(t_e = p_i + p_j) \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \mathcal{N}(p_j | \mu_j, \beta_j^2 + \sigma_j^2) dp_i dp_j \\
 &= \int \mathcal{N}(p_i | \mu_*, \sigma_*^2) \overbrace{\mathcal{N}(t_e | \mu_i + \mu_j, 2 * \beta^2 + \sigma_i^2 + \sigma_j^2)}^{\text{Scalar independent of } p_i} dp_i \\
 &= \mathcal{N}(t_e | \mu_i + \mu_j, 2\beta^2 + \sigma_i^2 + \sigma_j^2)
 \end{aligned} \tag{2.24}$$

Por inducción se demuestra que en un equipo con n jugadores, la probabilidad de un dado rendimiento de equipo viene dado por,

$$p(t_e | A_e) = \mathcal{N} \left(t \mid \sum_{i \in A_e} \mu_i, \sum_{i \in A_e} (\beta^2 + \sigma_i^2) \right) \tag{2.25}$$

Donde la probabilidad de rendimiento de equipo queda como una distribución gaussiana con media igual a la suma de las medias de habilidad de los integrantes y con una varianza igual a la suma de las varianzas individuales de la probabilidad de rendimiento mostrada en la ecuación 2.21.

Diferencia La diferencias en los rendimientos de equipos es la que determina el resultado del juego, definida como $d_{ab} = t_a - t_b$. Donde d_{ab} representa la diferencia entre el equipo a y el equipo b . De la misma manera como se mostró en la figura 2.5, las líneas de contorno de diferencia de rendimiento son líneas paralelas a la diagonal de diferencia cero. La probabilidad de una dada diferencia de rendimiento d_{ab} viene dada por,

$$p(d_{ab}|A_a, A_b) = \iint \mathbb{I}(d_{ab} = t_a - t_b) \cdot \mathcal{N}(t_a | \sum_{i \in A_a} \mu_i, \sum_{i \in A_a} \beta^2 + \sigma_i^2) \cdot \mathcal{N}(t_b | \sum_{i \in A_b} \mu_i, \sum_{i \in A_b} (\beta^2 + \sigma_i^2)) dt_a dt_b \quad (2.26)$$

donde A_a representa al conjunto de agentes del equipo a y A_b representa al conjunto de agentes del equipo b . Se puede mostrar que la probabilidad de una dada diferencia de rendimiento d_{ab} viene dada como,

$$p(d_{ab}|A_a, A_b) = \mathcal{N}\left(d_{ab} \mid \underbrace{\sum_{i \in A_a} \mu_i - \sum_{i \in A_b} \mu_i}_{\text{Expected difference } (\delta)}, \underbrace{\sum_{i \in A_a \cup A_b} \beta^2 + \sigma_i^2}_{\text{Total variance } (\vartheta)}\right) = \mathcal{N}(d_{ab} \mid \delta, \vartheta) \quad (2.27)$$

Nuevamente se obtiene una distribución gaussiana. En este caso, su media es la diferencia entre la media de la probabilidad de rendimiento del equipo ganador y la de su contrincante. La varianza queda como la suma de las varianzas de las probabilidades de rendimiento de los equipos. Se recuerda que el método **TrueSkill** organiza a los equipos por ganadores. Es decir, en este caso el equipo a fue el ganador.

Resultado El resultado de un evento es la variable observable que es la que nos termina aportando la información que alimenta el modelo. Una victoria del equipo a sobre otro equipo b se modela como:

$$r_{ab} = \begin{cases} 0, & \text{if } d_{ab} > 0 \\ 1, & \text{otherwise} \end{cases} \quad (2.28)$$

donde r_{ab} es el resultado de la partida del equipo a versus el equipo b . Tiene el valor 0, si el equipo a es el ganador, y 1 en caso contrario. Luego, la probabilidad de victoria de un equipo sobre el otro, se computa como:

$$P(r_{ab} = 0|A_a, A_b) = p(d_{ab} > 0|A_a, A_b) = \Phi(0|\delta, \vartheta^2) \quad (2.29)$$

donde nuevamente Φ es la función de la distribución acumulada de la distribución normal con media δ y desviación estándar ϑ . En el caso de múltiples equipos, el resultado observado de un juego, se modela con un vector de equipos ordenados, o , tal que $t_{o_1} > \dots > t_{o_{|A|}}$, siendo el ganador el de mayor valor.

Posterior En resumen, el modelo de **TrueSkill** puede ser representado por una red gráfica como se muestra nuevamente en la figura 2.12.

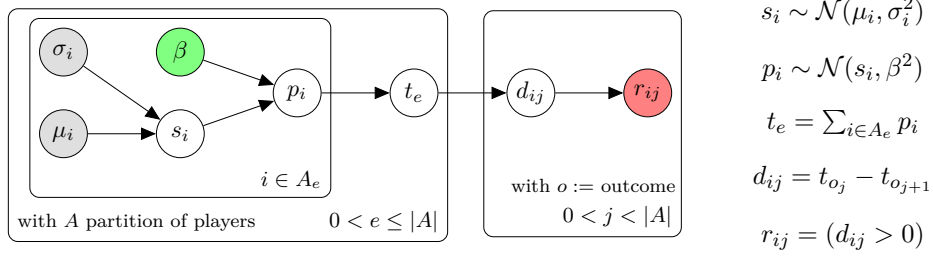


Figura 2.12: Representación esquemática del modelo generativo **TrueSkill**. Las habilidades ocultas causan los resultados observables a través de la diferencia de rendimientos ocultos. En cada evento, gana el agente quien haya obtenido mayor rendimiento, $r = p_i > p_j$. Esta representación gráfica define una distribución de probabilidad conjunta. Las variables observables se pintan de rojo, la ocultas en blanco, los parámetros constantes en verde y los parámetros de la distribución *prior* en gris.

La ecuación 2.1 permite actualizar nuestras creencia sobre ciertas hipótesis, dado un modelo y los datos.

$$\underbrace{p(\overbrace{\text{Skills}}^{\text{Hidden}} | \overbrace{\text{Result, Model}}^{\text{Observed}})}_{\text{Posterior}} = \frac{\overbrace{P(\text{Result} | \text{Skills, Model})}^{\text{likelihood}} \overbrace{p(\text{Skills})}^{\text{Prior}}}{\underbrace{P(\text{Result} | \text{Mode})}_{\text{Evidence}}} \quad (2.1)$$

En nuestro caso, las hipótesis son las habilidades de las personas (*Skills*), los datos son los resultados (*Result*) y el modelo está definido por la figura 2.7. Con una distribución de probabilidad que represente nuestras creencias a priori (*prior*) sobre las habilidad de las personas, podemos obtener la distribución de creencias a posteriori (*posterior*) exacta usando tan solo las reglas de la suma y el producto, mediante el algoritmo *sum-product* explicado en la sección anterior 2.1.4 ([Pasaje de mensajes](#)) sobre el gráfico de factores (figura 2.7). En el caso de dos jugadores, la *posterior* exacta queda como:

$$\underbrace{p(s_i, r_{ij} | s_j)}_{\text{Posterior}} \propto \underbrace{[1 - \Phi(0 | \overbrace{s_i - \mu_j}^{\delta(s_i)}, \overbrace{2\beta^2 + \sigma_j^2}^{\vartheta_i^2})]}_{\text{Likelihood}} \underbrace{\mathcal{N}(s_i | \mu_i, \sigma_i^2)}_{\text{Prior}} \quad (2.30)$$

donde el *posterior* se encuentra sin normalizar, para ello se lo divide por $P(r_{ij})$. La única variable libre es la hipótesis s_i . La incertidumbre posterior de cada hipótesis es proporcional al producto entre la propia incertidumbre previa y su *likelihood*. Se puede notar que el *likelihood* es solo la predicción previa del resultado ganador observado, realizada con todas las hipótesis de habilidad del oponente, suponiendo s_i como la verdadera habilidad del agente. Eso es lo que computa la densidad acumulada de todos los valores positivos de la distribución gaussiana centrada en la diferencia entre la hipótesis s_i y la media estimada del oponente s_j , con una varianza que incluye todas las incertezas excepto la de s_i , ϑ_i^2 .

Mensaje aproximado El modelo TrueSkill utiliza el método *expectation propagation* para realizar las aproximaciones necesarias. En particular, la forma que realiza esta aproximación, es mediante la minimización de la divergencia de **Kullback-Leibler** que es una medida de que tan bien es aproximada una distribución $q(x)$, en comparación con otra distribución, $p(x)$. Si $p(x)$ es una gaussiana truncada, esta métrica se minimiza al elegir una gaussiana, $q(x)$, con la misma media y la varianza que $p(x)$. En este caso de

expectation propagation, la forma de calcular la media y varianza de una distribución gaussiana truncada se deja en el apéndice B.1.3.

El mensaje $p(d)$ es el mensaje de subida de los nodos diferencias (en caso de un jugador en contra de otro, hay un único nodo diferencia). Existen $n - 1$ nodos de diferencia, siendo n la cantidad de equipos. Para realizar esta aproximación, primero se calcula la marginal del nodo diferencia, teniendo así que calcular todos los mensajes entrantes a este nodo. La marginalización de éste queda una gaussiana truncada.

$$p(d) = \mathcal{N}(d|\delta, \vartheta) \mathbb{I}(d > 0) \quad (2.31)$$

Si suponemos que el jugador i ganó sobre j , estos parámetros quedan de la forma, $\delta = \mu_i - \mu_j$ y $\vartheta^2 = \sigma_i^2 + \sigma_j^2 + 2\beta^2$. La aproximación de $p(d)$ se la denota como $\hat{p}(d)$, definida como,

$$\hat{p}(d) = \mathcal{N}\left(d \mid \delta + \vartheta V_1(t), \vartheta^2(1 - W_1(t))\right) \quad (2.32)$$

Las funciones $V_1(t)$ y $W_1(t)$, se explican en el apéndice B.1.3. Éstas son combinaciones de distribuciones normales y acumuladas, que aparecen al calcular el primer y segundo momento de la distribución $p(d)$. El parámetro t , está definido como $t = (\delta - x_1)/\vartheta$, con x_1 el valor donde se encuentra truncada la gaussiana. Teniendo la marginal aproximada, se extrae la información del mensaje que no es de interés, es decir el mensaje del factor diferencia f_d al nodo diferencia, d , de modo que nos quede el mensaje de subida aproximado. Ésto se realiza simplemente dividiendo la marginal por este mensaje.

$$m_{d \rightarrow t_d} \propto \widehat{m_{d \rightarrow t_d}} = \frac{\hat{p}(d)}{m_{f_d \rightarrow d}} = \mathcal{N}(d, \delta_{\div}, \vartheta_{\div}^2) \quad (2.33)$$

donde δ_{\div} y ϑ_{\div} son la media y varianza que mejor aproximan la gaussiana truncada. Podemos ver que la aproximación se hace de manera local en el nodo de factor, y en una forma independiente a la estructura restante del grafo. Con este mensaje, se puede continuar con la marginalización de los nodos de interés, para poder calcular los *posteriors* de los jugadores preservando la forma gaussiana.

Múltiples juegos Como ya mencionamos, TrueSkill utiliza el *posterior* de un jugador como *prior* de su siguiente juego. Para ello se usa el enfoque de inferencia aproximada conocida como **online learning** (llamado algunas veces como *filtering*), en donde cada distribución de habilidad de jugador, obtiene una actualización solo cuando se obtiene un nuevo resultado de juego en el cual involucre éste jugador. Cuando un agente juega una nueva partida, se realiza inferencia usando esta actual distribución de habilidad gaussiana como *prior*, y la distribución a *posterior* resultante es luego guardada para ser el *prior* del próximo juego del agente en cuestión. Cada juego individual es entonces descrito por el grafo de la forma mostrada de la figura 2.7. Para TrueSkill, el *posterior* con distribución gaussiana calculado en cualquier tiempo es,

$$\widehat{\text{Posterior}}_n \propto \widehat{\text{Likelihood}}_n \overbrace{\widehat{\text{Likelihood}}_{n-1} \dots \widehat{\text{Likelihood}}_2 \widehat{\text{Likelihood}}_1 \widehat{\text{Prior}}_1}^{\widehat{\text{Posterior}}_{n-1} \text{ as Prior}_n} \quad (2.34)$$

$\widehat{\text{Posterior}}_1 \text{ as Prior}_2$

donde $\widehat{\text{Posterior}}_n$ y $\widehat{\text{Likelihood}}_n$ representan las aproximaciones inducidas por la ecuación 2.33 en el n -ésimo evento. Si consideramos el *likelihood* como un filtro del *prior*, cada *posterior* puede ser visto como una acumulación de todos los filtros anteriores. De esta forma, la información se propaga de estimaciones pasadas hacia futuras. Debido a que las habilidades son variables dinámicas es importante agregar alguna incertidumbre γ en cada paso.

Ruido Por como esta formado el método, al utilizar los *posteriors* como *priors*, estaríamos haciendo la suposición de que cada jugador tiene un único valor de habilidad. Esto es un error, ya que un jugador puede evolucionar (i.e. mejorar su habilidad) en el tiempo, contradiciendo esta suposición. Para corregir esta situación, se utiliza la suposición de que cada jugador tiene un valor de habilidad, representado por una variable continua, dado por su valor de habilidad en su juego previo junto con algún tipo de cambio en la habilidad que tenga una media nula con una distribución en forma de campana, en particular gaussiana. Es decir se adiciona una constante de ruido γ al *posterior*. Al agregarle una incertidumbre a la habilidad del jugador, nos permite evitar valores de incerteza muy chicas, que terminarían impidiendo una correcta actualización de la media.

El valor de γ tiene que ser lo suficientemente grande para que pueda seguir a los cambios reales (ocultos) de la habilidad del jugador, pero no tanto como para que haga oscilar la habilidad en cada evento. La elección de este valor se discutirá en la sección de validación.

$$\text{Posterior} = \hat{p}(s_{i_n+1}) = \mathcal{N}(s_{i_n+1} | \mu_{i_n}, \sigma_{i_n}^2 + \gamma^2) \quad (2.35)$$

Debido a que el enfoque de filtrado no surge de ningún modelo probabilístico, sufre de una serie de problemas. Sus debilidades están relacionadas con el hecho de que la información propaga en un solo sentido por el sistema. El inicio de toda secuencia de estimaciones siempre tiene alta incertidumbre. Pero también sufre de desacoplamientos “temporales” y “espaciales”. A pesar de que la diferencia relativa entre estimaciones contemporáneas al interior de comunidades bien conectadas puedan ser correctas, las estimaciones separadas en el tiempo y entre comunidades poco conectadas suelen ser incorrectas. A fin de obtener buenas estimaciones iniciales y garantizar compatibilidad temporal y espacial, necesitamos un modelo causal del proceso temporal que vincule todas las actividades históricas. Este enfoque, conocido como *smoothing*, es el implementado por **TrueSkill Through Time** [11]. Es difícil imaginar la compleja estructura que pueden adquirir este tipo de modelos gráficos, cualquier videojuego requerirá típicamente una red de millones de eventos. Es importante notar que, cuando se hace *online learning*, el orden en el que se infiere las partidas toma una importancia significativa y puede influir a la evaluación de habilidad. Este tema se verá más adelante cuando se vea **TrueSkill Through Time**.

Evidencia Teniendo todas las herramientas, podemos ahora explorar un poco más el mecanismo de evaluación presentado inicialmente en la introducción teórica. Lo que nos interesa es utilizar la regla de Bayes para poder saber la probabilidad de que suceda un resultado de un juego dado nuestros *priors* (los cuales fueron inferidos en partidas previas). Para ello recordemos la ecuación **teorema de Bayes** que incluye la que queremos obtener como dividendo. Como esta probabilidad hace que nuestro *posterior* esté normalizado, si integramos el divisor nos dará exactamente lo que estamos buscando, $P(r_{ij})$. Podemos ver en la figura 2.1, la distribución *prior* y *likelihood* para un caso genérico, calculadas de manera exacta, junto con su *posterior* no normalizado, como vimos en la ecuación 2.30.

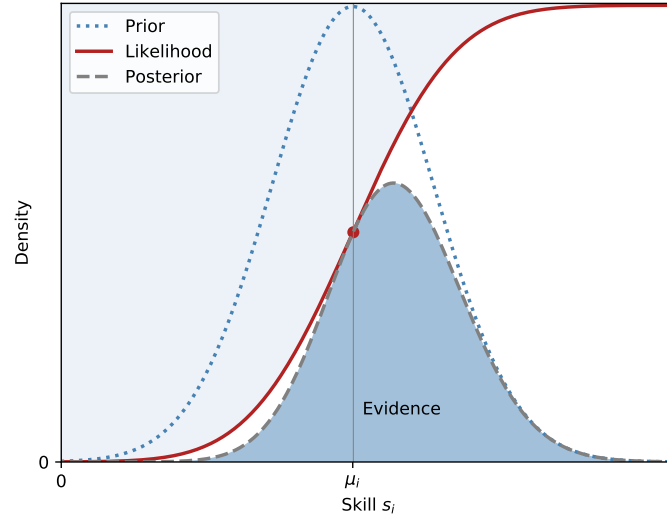


Figura 2.13: Actualización para el caso ganador usando *Belief propagation*. El *posterior* se obtiene con el producto del *prior* (distribución gaussiana) y el *likelihood* (función acumulada de la distribución gaussiana). La sorpresa, definido como el complemento del *likelihood*, funciona como filtro para el *prior*. La evidencia es la integración del *posterior*. Las distribuciones no están necesariamente en la misma escala: el *prior* integra 1, mientras que el *likelihood* tiene un rango entre 0 y 1.

El *posterior* recibe la densidad del *prior* no filtrado por el *likelihood*. En la región con hipotéticos valores grandes de habilidad, donde un resultado ganador prácticamente no genera sorpresa, el *posterior* recibe casi toda la densidad del *prior*. Es importante enfatizar que el *posterior*, a pesar de su apariencia, no es una distribución gaussiana, lo cual impediría usar la ecuación 2.30 de forma iterativa. Sin embargo, debido a la forma de esta *posterior* exacta, una gaussiana parece ser una buena aproximación como ya se vio en secciones anteriores. Ésto permite evitar los costos computacionales de diversos métodos para aproximar distribuciones.

El concepto de evidencia se mantiene para el caso aproximado y para el caso de multi-equipos que se verá en la siguiente sección. La forma practica de realizar ésto, es aprovechar que los equipos ya se encuentran organizados por el método, y de esa forma se puede estudiar la distribución diferencia, la cual tiene toda la información necesaria. Esta distribución, al haber utilizado la aproximación, tiene una forma funcional gaussiana, centrada en la diferencia. La integral de $-\infty$ a cero representa la probabilidad que tiene de ganar el jugador que en realidad perdió y la integral de cero a ∞ , representa la probabilidad de ganar que tiene el jugador ganador. En caso de múltiples equipos se genera esto para cada pares de diferencias, es decir para cada nodo de diferencia. Realizando de forma análoga, se calcula la acumulada de esta distribución hasta cero, y como en general interesa la probabilidad del ganador restamos esta acumulada a 1. Esto se puede observar en la figura 2.14

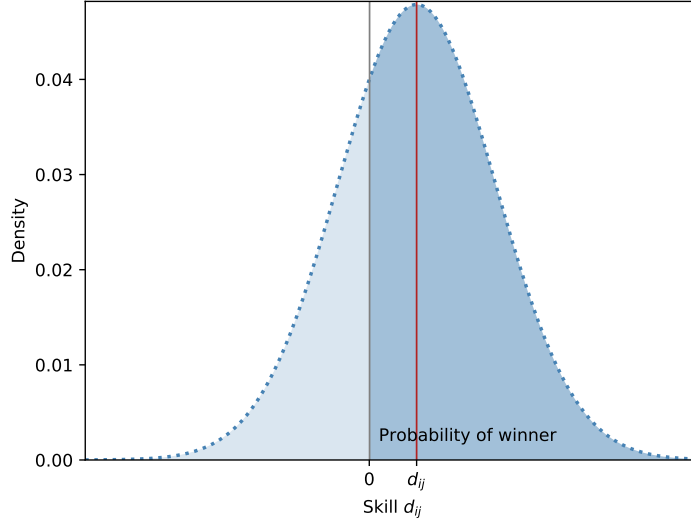


Figura 2.14: Distribución gaussiana de la diferencia. La integral hasta cero da la probabilidad de que el jugador perdedor gane.

Guardando este valor juego a juego, se multiplican las evidencias para así tener una medida de nuestro modelo. En nuestro caso utilizamos la evidencia logarítmica negativa y promediada. Si llamamos a la evidencia total como **Evidencia** y a la evidencia del juego i , como $ev_i = (1 - \Phi(0|\delta_i, \hat{v}_i^2))$, se tiene:

$$\text{Evidencia} = -\frac{\log(\prod_{i=1}^N ev_i)}{N} \quad (2.36)$$

De esta forma obtenemos un valor que representa el grado de coincidencia entre nuestro modelo y los resultados observados.

2.2.4 Full TrueSkill

Como fue mencionado, **TrueSkill** permite la implementación de juegos de múltiples equipos, al cual denominaremos como **Full TrueSkill**. A pesar de no ser usado en esta tesis, se considera de ayuda entender su funcionamiento debido al parecido en la implementación de este con el método de interés, **TTT**. En esta sección veremos cómo encontrar la mejor aproximación al *posterior* exacto en casos en los que tenemos más de dos equipos. Supongamos que n agentes, organizados en k equipos $\{1, \dots, k\}$, participan de un evento. La asignación de equipos se representa como una partición del conjunto de agentes, A , en k subconjuntos disjuntos, A_i . Gracias a la transitividad de los resultados, es suficiente con evaluar $k - 1$ distribuciones de diferencia d_i entre equipos consecutivos en el ranking. Para ese propósito definimos el orden de equipos que surge del resultado observado, $o := (o_1, \dots, o_k)$, donde o_1 indica el equipo ganador, y en general $o_i = e$ indica que el equipo e quedó ubicado en la posición i . La figura 2.15, muestra a modo de ejemplo el caso de tres equipos. Se muestra solo hasta el factor de equipo debido a que las operaciones en los demás nodos es análogo a **TrueSkill** con solo dos equipos. Es importante volver a resaltar que el método organiza a los equipos en orden de ganador a perdedor

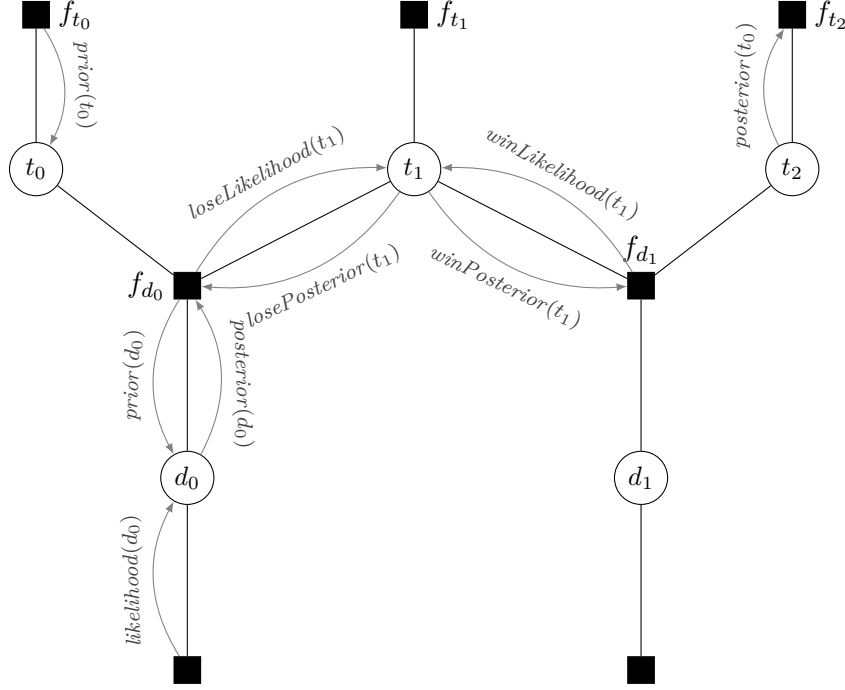


Figura 2.15: Grafo bipartito de la factorización (*factor graph*) del modelo TrueSkill para el caso de multi-equipos. Se recortaron algunos nodos para una mejor visualización de los cambios relevantes al caso de TrueSkill explicado anteriormente.

Los nombres fueron elegidos para simplificar la notación. Podemos observar que una diferencia importante es que, ahora, los nodos de la variable de rendimiento de equipo, t_i , puede estar conectada a dos nodos de factores de diferencia, f_{d_i} . Esto hace que haya una dependencia mutua entre las marginales de la diferencia de rendimiento de equipos, $p(d_i)$, haciendo imposible realizar una inferencia directa sin bucle. La idea básica es actualizar rápidamente, tanto para adelante como para atrás, todos los mensajes en el camino más corto entre cualquiera dos marginales $p(d_i)$ hasta que converja. Donde “para adelante” se entiende como los mensajes que van del nodo f_{d_0} hacia f_{d_n} , donde n es el último equipo y el sentido “para atrás” se entiende para la dirección contraria. Los mensajes que todavía no están definidos, por ejemplo $winLikelihood(t_1)$ en la primer pasada hacia adelante, tienen que ser reemplazados por una forma neutral, como puede ser un escalar unidad o una distribución gaussiana con varianza infinita. Si el mensaje ya había sido definido, entonces usamos la última definición posible.

En el primer ciclo, se calcula el mensaje $prior(d_0)$. Para ello se calculan los mensajes que le llegan al factor f_{d_0} , $m_{t_0 \rightarrow f_{d_0}}(t_0)$ y $m_{t_1 \rightarrow f_{d_0}}(t_1)$. El primero de estos mensajes se realiza igual que en TrueSkill, la diferencia radica en el segundo. En éste tenemos que calcular primero los mensajes que le llegan al nodo t_1 . En el tenemos el mensaje análogo a t_0 pero además tenemos el mensaje que viene del nodo de factor f_{d_1} , el cual todavía no fue calculado. Este ultimo mensaje se envía con una forma neutra. Con ésto logramos calcular la marginal $p(d_0)$ con la aproximación igual que en la sección anterior,

$$\hat{p}(d_i) = prior(d_i) \widehat{likelihood}(d_i) \quad (2.37)$$

$$\widehat{likelihood}(d_i) = \frac{approx(prior(d_i) \cdot likelihood(d_i))}{prior(d_i)} \quad (2.38)$$

De esta forma, subimos con el mensaje aproximado, y se lo pasamos al nodo de la izquierda (t_1) como el mensaje $loseLikelihood(t_{i+1})$. Teniendo esta información procedemos de la misma forma a calcular la marginal $p(d_{i+1})$. Se prosigue hasta llegar al último nodo de diferencia, y se repite el proceso invirtiendo el sentido. Esta vez, luego de cada marginal se entrega el mensaje $winLikelihood(t_i)$ al nodo de la derecha. Se prosigue calculando la marginal $p(d_i)$, repitiendo el proceso hasta llegar al primer nodo $p(d_0)$. Esto se reitera hasta que no haya cambios relevantes en los mensajes. De esta forma se asegura que toda la información del evento se propague a todos los *posteriors* de los agentes en cuestión. De esta forma podemos generalizar todos los mensajes y marginales de interés. En el apéndice B.2.1 (Full TrueSkill)

2.2.5 TrueSkill Through Time

La novedad del método TrueSkill Through Time (TTT) [11], consiste en la propagación temporal de la información. En el método anterior, a medida que obtenemos un nuevo observable, se actualiza el conocimiento de la habilidad de los jugadores en cuestión, utilizado esta información para un evento futuro. En cambio TTT no solo realiza esto, si no que a su vez utiliza esta nueva información, para propagarla en sentido contrario y así actualizar la información de eventos pasados. De esta forma a medida que obtenemos más eventos de un jugador, podemos estimar de manera más certera la habilidad del jugador que tuvo en cada evento, incluso en la primer partida observada. En TrueSkill, esta primer partida posee mucha incertidumbre debido a que es la única información hasta ese momento. Ésto también implica que este método logra obtener una habilidad certera de un jugador con una menor cantidad de eventos del mismo. Por como esta formado el método, en la primer pasada de la primer iteración, se obtiene TrueSkill, pudiendo guardar esta información de forma separada en caso de requerirse.

En esta sección vamos a ver cómo se obtiene la mejor aproximación al posterior exacto en una historia de eventos en la que n agentes compiten durante T pasos temporales o *batches* (e.g. día, semana, mes, año). Al interior de cada paso temporal t un agente i puede participar en K eventos, K_{i_t} que resulta en una única habilidad s_i^t . La inferencia entre un lote temporal depende del orden aleatorio elegido para la actualización. De todas formas el resultado sera independiente del orden de los juegos dentro de un lote temporal. En la figura 2.16 se muestra una representación esquemática del grafico de factorización de una historia de eventos para una variable de habilidad temporal s_i^t .

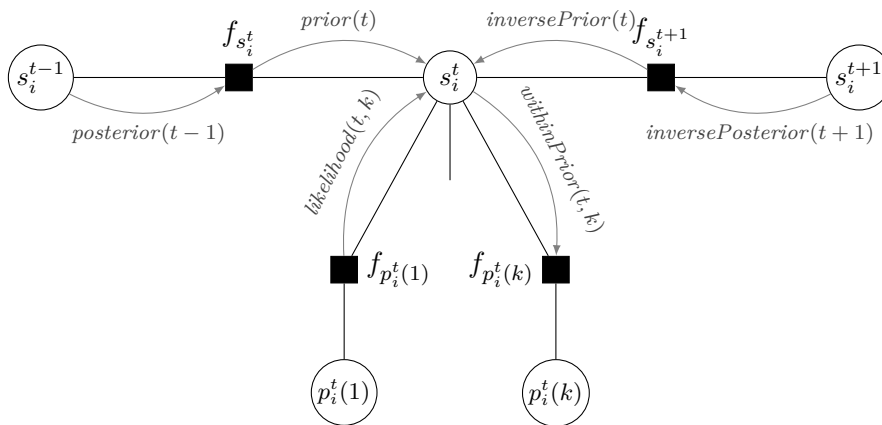


Figura 2.16: Representación del grafo bipartito de la factorización (*factor graph*) del modelo TTT para el caso de la habilidad de un jugador i .

Debido que en cada lote hay una única habilidad por jugador, cada nodo s_i^t representa la información de un mismo jugador dentro de un lote. Nuevamente los nombres fueron elegidos solamente para simplificar la notación. Las flechas representan los mensajes computados por el algoritmo de *sum-product*. Gracias al mismo, sabemos que la distribución marginal de cualquier variable es el producto de los mensajes que esta variable recibe de sus vecinos,

$$p(s_i^t) = \prod_{h \in n(s_i^t)} m_{h \rightarrow s_i^t}(s_i^t) \quad (2.39)$$

Remplazando los mensajes por los nombres seleccionados, esta distribución se puede expresar como,

$$p(s_i^t) = \text{prior}_i(t) \cdot \text{inversePrior}_i(t) \cdot \prod_{k=1}^{K_i^t} \text{likelihood}_i(t, k) \quad (2.40)$$

Los mensajes *prior* e *inversePrior* son las estimaciones de habilidad vecinas, a las que se le agrega cierta incertidumbre γ por el paso temporal. Dentro de un lote temporal, no se adiciona ninguna constante de ruido, como se realiza en **TrueSkill**. Este parámetro se adiciona entre lote y lote.

$$\text{prior}_i(t) = \mathcal{N}(s_{it} | s_{it-1}, \gamma^2) \quad \text{inversePrior}_i(t) = \mathcal{N}(s_{it} | s_{it+1}, \gamma^2)$$

los *likelihoods* se computan siguiendo la sección 2.2.4 (**Full TrueSkill**), usando como prior toda la información salvo la de la partida de interés.

$$\text{withinPrior}_i(t, k) = \text{prior}_i(t) \cdot \text{inversePrior}_i(t) \cdot \prod_{\substack{q=1 \\ q \neq k}}^{K_{it}} \text{likelihood}_i(t, q) = \frac{p(s_i^t)}{\text{likelihood}_i(t, k)} \quad (2.41)$$

Existe una dependencia mutua entre los mensajes hacia adelante y hacia atrás, haciendo imposible una inferencia en una única iteración. La idea básica es la de actualizar repetidas veces los mensajes hacia adelante y hacia atrás hasta su convergencia, asegurándonos que el efecto de la actualización previa sea removida antes de que el nuevo efecto se agregue. Los mensajes que no están todavía definidos, por ejemplo el *inversePrior* en la primera pasada hacia adelante, es remplazado por una forma neutral como un escalar unidad o una distribución gaussiana con infinita varianza. Finalmente, los mensajes que la variable s_{it} envía al pasado y al futuro son,

$$\text{posterior}_i(t-1) = \frac{p(s_{it-1})}{\text{inversePrior}_i(t-1)} \quad , \quad \text{inversePosterior}_i(t+1) = \frac{p(s_{it+1})}{\text{prior}_i(t+1)}$$

En cada pasada hacia adelante, guardamos cada mensaje hacia adelante (i.e. $\text{prior}_i(t+1)$) y en cada pasada hacia atrás computamos el mensaje hacia atrás (i.e. $\text{inversePrior}_i(t-1)$). De esta forma, *TTT* logra mejorar las estimaciones de las habilidades en relación a su antecesor con una menor cantidad de eventos. En el apéndice B.2.2 (**TrueSkill Through Time**), se pueden encontrar todas las definiciones de los diferentes mensajes de la representación gráfica 2.16.

En el caso de esta tesis, se utiliza TTT con un único lote temporal por partida. De esta forma el análisis se centra en la caracterización e impacto de las piedras de *handicap* y no en el impacto asociado con los posibles tamaños de los lotes.

2.2.6 Visualización

Para finalizar este capítulo veamos, entonces, un ejemplo sobre qué pasa con las distribuciones de habilidades luego de aplicar *expectation propagation*. Usaremos el ejemplo de un juego de go, donde el evento consiste del jugador blanco enfrentándose al negro. Consideremos que el jugador negro es un jugador nuevo para el método, en el cual cuenta con mucha varianza, mientras que el jugador blanco cuenta con una habilidad con más información, teniendo una menor varianza y además consideremos que tiene una media más grande que el jugador negro. En la figura 2.17, podemos ver el *prior* y *posterior* de la distribución de habilidad de los jugadores para los dos casos posibles. En la figura 2.17(a) tenemos el caso donde el jugador negro es el que gana, mientras que en la figura 2.17(b), la gana el jugador blanco.

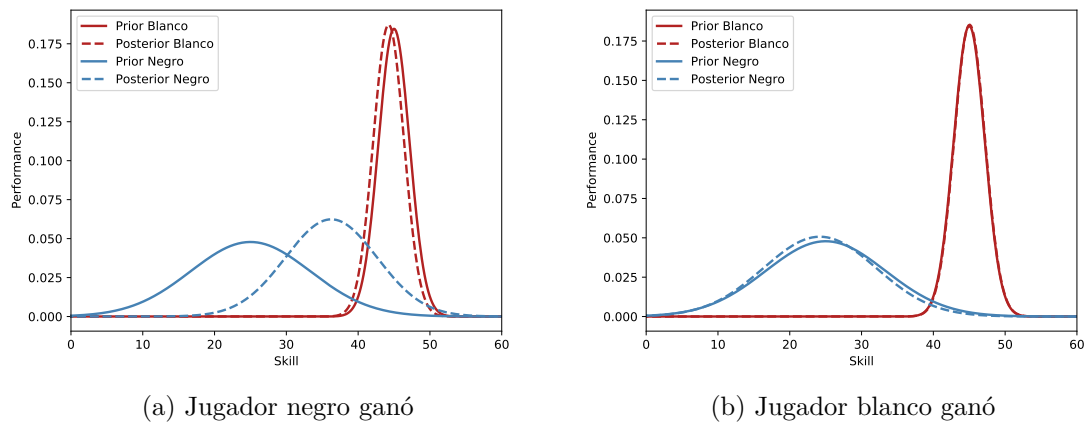


Figura 2.17: Descripción grafica de las habilidades de los jugadores involucrados antes y después de actualizar mediante *TrueSkill*. En color rojo se muestra la habilidad del jugador blanco, mientras que en azul la del jugador negro. Las curvas continuas representan los *priors* mientras que las curvas punteadas los *posteriors*. En figura (a) se observa el caso en donde el jugador negro es el ganador, mientras que en la figura (b) se muestra el caso en donde es el perdedor.

En ambos casos podemos ver un aumento en la media del jugador ganador, mientras que la del perdedor se reduce. Lo interesante es que estos aumentos o reducciones varían en cada caso. En el primero, representado en la figura 2.17(a), se puede observar un gran aumento de la media del jugador negro, mientras que la media del jugador blanco decrece levemente. Esta diferencia se debe a la gran certeza que hay acerca de la habilidad del jugador blanco en comparación con la del jugador negro. Podemos decir que estamos usando la habilidad del jugador blanco para estimar la del jugador negro. Esto es una diferencia crucial a la de *Elo*, en donde el cambio en habilidad estimado es simétrico para ambos jugadores. En la figura 2.17(a) también se observa que la desviación estándar de la distribución de habilidad del jugador negro decreció. Esto es debido al resultado de este juego nos permite aprender algo acerca de su habilidad y, por eso, decrece el grado de incertidumbre.

Alternativamente, si el jugador blanco es el ganador, se tiene el resultado que se muestra en la figura 2.17(b). Este resultado es menos sorprendente, debido que teníamos la creencia de que el jugador blanco era más fuerte y terminó ganando. Intuitivamente, esperaríamos que el ajuste en la distribución de habilidad sea leve. Podemos ver que el cambio de la media de la distribución resulta muy leve en comparación al caso de la figura 2.17(a). Nuevamente, la desviación estándar del jugador negro decreció, en menor cantidad, reflejando la reducción en su incertidumbre debido a la incorporación de nueva información del agente.

Debido a que las habilidades en el modelo de **TrueSkill** se actualizan dependiendo de la varianza de la distribución de habilidad del agente, **TrueSkill** logra hacer cambios grandes en las distribuciones de nuevos jugadores. Esto sucede de forma automáticamente como consecuencia de correr inferencia en nuestro modelo. A diferencia de las actualizaciones de **Elo** que están gobernadas por el parámetro de actualización K de la ecuación 2.16. Al seguir la información de incertidumbre en éste modelo, evitamos la necesidad de cambiar ad-hoc el parámetro.

Capítulo 3

Validación

Una de las etapas fundamentales en todo análisis y modelado es el testeo del software utilizado. A su vez, en nuestro caso, esta etapa también sirve para obtener un mayor entendimiento del modelo y sus alcances. Es por ello que este capítulo se dividió en dos partes. En la primer sección, [Validación](#), se realizaron diversas simulaciones con el fin de validar y probar el paquete de `Python` con la implementación de `TrueSkill` y `TTT`, utilizado en esta tesis y desarrollado en el laboratorio. La siguiente sección, [Análisis del modelo](#), se enfocó en diversas pruebas variando los parámetros del modelo como también las relaciones entre éstos, con el fin de obtener la experiencia necesaria para elegir los parámetros que se utilizarán en los análisis posteriores.

3.1 Validación

En el caso de `TrueSkill`, se corroboraron los resultados obtenidos con nuestra implementación en diferentes escenarios, comparándolos con el paquete ya existente de `Python` de `TrueSkill`. En cuanto a `TrueSkill Through Time`, al no contar con un paquete para poder comparar, se realizaron diversas pruebas para validar su resultado. La primer prueba sencilla, fue recrear el caso en el cual tres jugadores, juegan una partida entre cada uno, ganando una y perdiendo la otra. Se supone que, si esto se realiza en un intervalo temporal corto, los jugadores deberían tener la misma habilidad con una misma incertidumbre dentro del modelo. Para el caso de `TrueSkill`, se obtendrían los *posteriors* en cada partida como se muestran en la tabla [3.1](#).

Event	Winner	Looser	Posterior winner	Posterior looser	Final posterior (TTT)
a vs b	a	b	28.34	21.66	25.00
b vs c	b	c	25.06	20.08	25.00
c vs a	c	a	25.22	22.31	25.00

Cuadro 3.1: *Posteriors* de tres partidas utilizando `TrueSkill` con jugadores inicializados con una media de 25 y una incertidumbre de 6. Se recuadra en celeste el *posterior* final de cada jugador.

Se observa en la tabla [3.1](#) que, a pesar de que cada jugador tiene la misma cantidad de partidas ganadas y perdidas, al importar el orden de realización de las partidas, los *posteriors* finales (recuadro celeste) son distintos. Por otro lado, cuando utilizamos nuestra

implementación de **TrueSkill Through Time**, los jugadores finalizan con una misma *posterior*, misma media y misma desviación estándar.

El siguiente ejemplo corresponde con el caso en el cual solo se consideran dos jugadores que se enfrentan una cantidad fija de veces. Luego, se generan diferentes escenarios aumentando la cantidad de partidas ganadas por cada jugador, pero manteniendo fijas la cantidad de derrotas. De esta forma, se varía la proporción de partidas ganadas y, por ende, la probabilidad de ganar que tiene un jugador con respecto a su competidor. Para cada uno de estos escenarios se estima la probabilidad de ganar de un jugador utilizando el modelo TTT. En la figura 3.1, se puede observar el valor esperado y la serie TTT_1 que corresponde con la estimación predicha para las diferentes fracciones de partidas ganadas. Adicionalmente, se crearon otros dos escenarios de análisis basados en aumentar la cantidad de partidas manteniendo la proporción en cada caso: TTT_2 corresponde con duplicar las partidas utilizadas en TTT_1 , mientras que para TTT_4 se cuadruplica el tamaño de la base de partidas. Es decir, evento tras evento se lo considera reiteradas veces como si fueran eventos diferentes. De esta manera, se busca analizar si al contar con mayor información (número de partidas) se puede mejorar la predicción realizada.

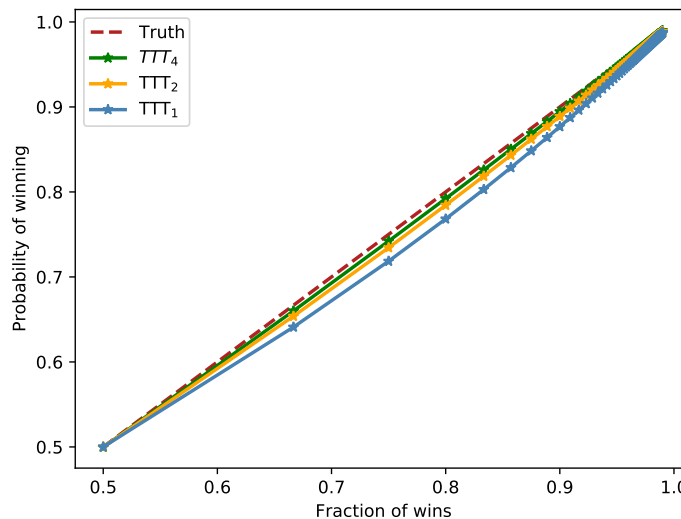


Figura 3.1: Probabilidad de ganar entre dos jugadores de estudio en función a la fracción de partidas ganadas por uno de ellos. Se muestran tres curvas con una cantidad creciente de partidas. Se duplica para la curva amarilla y se cuadruplica en la curva verde.

La figura 3.1 muestra que las tres series tienen una aproximación buena al valor esperado. A pesar de esto, se nota una leve diferencia entre ellas. Esta diferencia corrobora un hecho conocido en la estadística: mientras mayor información se tenga, más parecida es la estimación al valor real. En nuestro caso, esto se ve en que la serie TTT_2 aproxima mejor que la serie basada en la base de datos con la mitad las partidas (TTT_1), en tanto al contar con una base aún mayor (TTT_4), la estimación mejora más todavía acercándose al valor esperado en casi todo el rango analizado.

Otro caso interesante es el de un jugador del cual conocemos su evolución de habilidad y lo hacemos jugar en una comunidad de tamaño teóricamente infinito. Es decir, partida tras partida el jugador tiene un contrincante diferente sin que repita en ninguna partida en el futuro. Además, se conoce con precisión la habilidad del resto de los jugadores (i.e. tienen una varianza baja), que tienen una media de habilidad con un valor elegido mediante

una distribución gaussiana centrada en la media del agente objetivo al momento de jugar cada partida. Dicha elección se utiliza para recrear un escenario en donde un jugador se enfrenta a otro de un nivel similar. Se consideran 1000 partidas del jugador, en las cuales se muestrean los rendimientos utilizando las distribuciones de habilidad de ambos jugadores, siendo el ganador el que obtuvo el mayor valor. La evolución de la habilidad del jugador sigue un ley de potencia con exponente fijo (0,13) y como base el número de partidas jugadas. Este escenario se eligió para enfocarse en cómo se captura el cambio de la habilidad del jugador sin importar lo que sucede con la comunidad en la que está inserto.

Para poder obtener la mejor estimación, se requiere minimizar la evidencia del modelo y, para ello, encontrar el parámetro de ruido (γ) que lo logre. Haciendo un barrido sobre γ , se puede visualizar la evidencia en función de este parámetro tal como se observa en la figura 3.2,

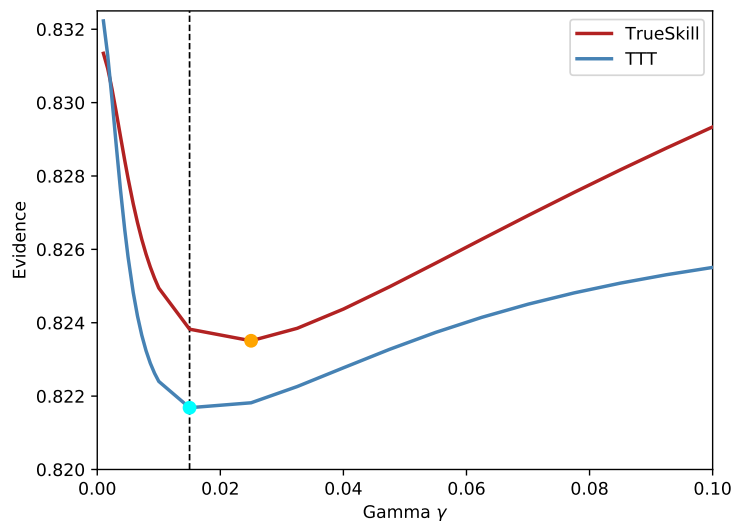


Figura 3.2: Evidencia en función del parámetro γ para los modelos de TrueSkill (curva roja) y TTT (curva azul). La línea vertical marca el γ donde la evidencia se minimiza para el método TTT y el punto naranja indica el mínimo para TrueSkill.

La figura 3.2 muestra que se encuentra un mínimo para γ en 0,015 para TTT y en 0,025 para TrueSkill, que consideraremos como *ideales*. Además, se puede ver cómo luego de que cada método llega a su mínimo, la pendiente ascendente que tiene TrueSkill es mayor que la de TTT, mostrando como este último es menos sensible ante este parámetro. Algo interesante es ver que el γ ideal en TrueSkill y en TTT no son iguales pero sí se encuentran cercanos. Esto es de ayuda ya que el primero es computacionalmente más económico de encontrar que el segundo. Un método para poder obtener el γ ideal consiste en hacer un barrido más burdo usando TrueSkill, para luego encontrar el γ ideal realizando un barrido fino con TTT, que es un método más caro. En la figura 3.3 se muestra la curva de aprendizaje real y diversas curvas estimadas para algunos valores del parámetro γ .

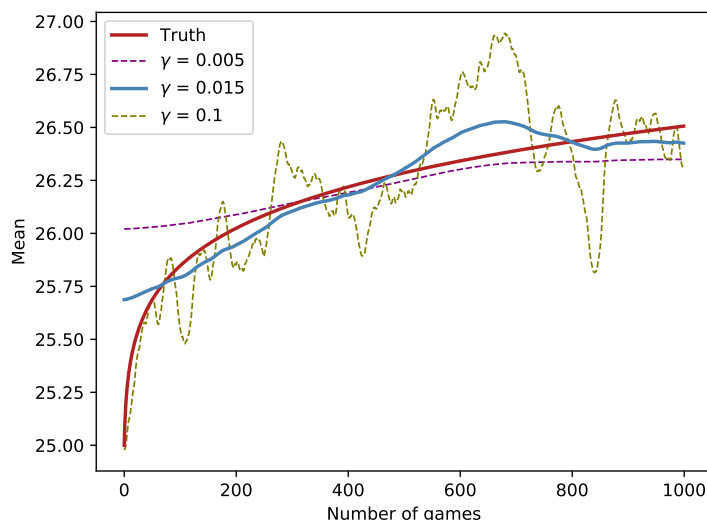


Figura 3.3: Estimación de la habilidad de un agente que evoluciona de acuerdo a una ley de potencia (curva roja) en función de la cantidad de partidas. Se muestran las diferentes curvas de aprendizaje estimadas con el método TTT y tres valores diferentes de γ . En azul se muestra la curva de la habilidad correspondiente con el γ ideal. Las otras dos curvas representan estimaciones que resultan muy rígidas o muy fluctuantes.

Se puede observar cómo un valor bajo de γ hace que el modelo no logre ser lo suficientemente rápido para actualizar en relación a los cambios de habilidad reales del jugador. Dicho de otra forma, no se le permite al sistema relajarse para poder llegar al valor deseado. En el caso opuesto, un valor de ruido grande, genera que partida tras partida (o lote tras lote) se agregue demasiada incertidumbre, no logrando seguir la habilidad real. Este ejemplo muestra la importancia y una forma de confirmación del procedimiento obtenido, mostrando cómo el caso de menor evidencia es el que mejor comportamiento reproduce.

■ 3.2 Análisis del modelo

Como vimos, *TrueSkill* modela a la habilidad de un agente mediante una distribución gaussiana la cual en cada evento presenta un rendimiento con una distribución similar de misma media pero con una varianza diferente la cual es constante para todos los jugadores. A su vez implica que el jugador o equipo con mayor rendimiento es el vencedor. Una vez convencidos de que el modelo y software presentan un comportamiento correcto, se realizó un estudio para comprender de mejor manera el modelo y sus parámetros. Por un lado se encuentra la media μ_0 y desviación estándar σ_0 con los cuales los nuevos jugadores se inicializan. Luego tenemos el parámetro de desviación estándar del rendimiento, β . Por último, el ruido (γ) que se agrega a los *posteriors*.

Para estas pruebas se realizó una simulación de una comunidad de 50 jugadores ficticios en las cuales cada uno tiene una habilidad aleatoria fija en un intervalo centrado en una media elegida de 25 con un ancho igual a dos veces la desviación estándar inicial, la cual fue de dos. El β elegido fue de uno. Del total de los jugadores se eligen dos para utilizarlos de fuente de estudio, eligiendo su habilidad con una diferencia de un β . De manera aleatoria se organizan 1000 partidas, eligiendo el ganador utilizando la distribución de rendimiento de

cada uno. El de mayor rendimiento es el ganador. En este caso sencillo donde los jugadores no varían su habilidad, la cantidad de partidas de 1000 con 50 jugadores se eligió para recrear la realidad donde la información no es infinita.

Teniendo la simulación, se utilizaron estos datos para realizar las estimaciones, realizando diferentes barridos para los parámetros. Una de las consideraciones es que en cada elección de parámetro dentro de un barrido, se mantienen todos los parámetros fijos, excepto en algunos casos, el γ . Este ultimo parámetro para cada caso se itera hasta encontrar el valor ideal, el cual minimiza la evidencia.

El primer caso, es un barrido para distintos valores de la media con la que se inicializan los jugadores (μ_0), se observa que la evidencia llegada con un mismo γ en cada caso es la misma. Para realizar el barrido, sin que TTT, converja a un único valor (debido a su propagación de información), cada iteración de este método se realizó una iteración por cantidad de partidas. Es decir, se fue agregando de a una partida y buscando la convergencia. Esto se lo conoce como *Online*. Podemos ver en la figura 3.4 algunas curvas de aprendizaje para un agente de estudio.

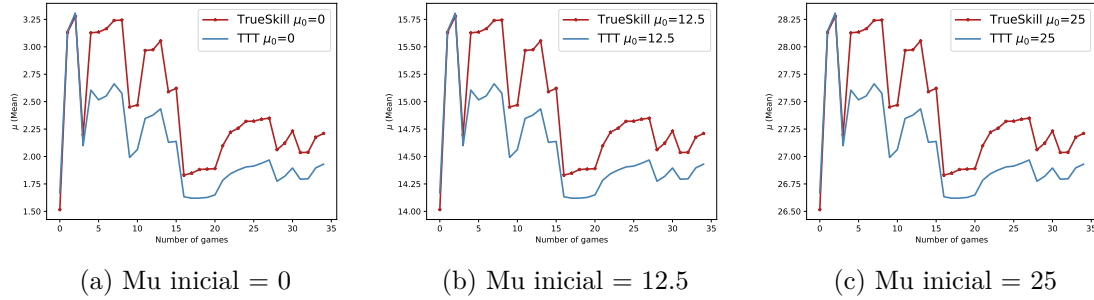


Figura 3.4: Media estimada para el agente de estudio en función a la cantidad de partidas. Se muestran tres casos de diferentes medias iniciales para los métodos TrueSkill y TTT. Este último método se uso de manera *online*. Se observa que en cada caso las curvas de un mismo modelo son iguales con una diferencia de sesgo.

En la figura 3.4 podemos observar cómo cada curva del mismo modelo tienen la misma forma con una diferencia de una constante. Justamente, lo que realiza el cambio de este parámetro, es nada más que agregarnos un sesgo a nuestra escala. Además, se encuentra que la diferencia entre todos los jugadores se mantiene constante. Es decir la escala relativa no varía o, dicho de otra manera, la probabilidad de ganar en cada evento no cambia. Esto se puede visualizar para el caso de los dos jugadores de estudio en la tabla 3.2

μ_0	Evidence TS	Evidence TTT	μ_1	μ_2	$\mu_1 - \mu_2$	Probability
0.0	0.53	0.47	1.98	0.98	1.00	0.74
12.5	0.53	0.47	14.48	13.48	1.00	0.74
25.0	0.53	0.47	26.98	25.98	1.00	0.74

Cuadro 3.2: Tabla de valores para los parámetros del modelo para el caso de tres habilidades iniciales diferentes. En estos casos, β es 1, el σ_0 inicial es 3 y el γ $1e - 07$.

Luego, se varió el parámetro σ_0 , donde en cada caso se iteró hasta encontrar el γ ideal. Intuitivamente, cuando σ_0 es muy chico en comparación a β , se están inicializando los jugadores nuevos (de los cuales no se conoce nada), con mucha información, dificultando

su actualización. En caso contrario, si fuera muy grande en relación a β , se inicializan con demasiado desconocimiento, produciendo un efecto similar. De todas formas, se puede observar en la figura 3.5, como en el caso de TTT, es mucho menos sensible a una desviación estándar grande comparado con el modelo básico.

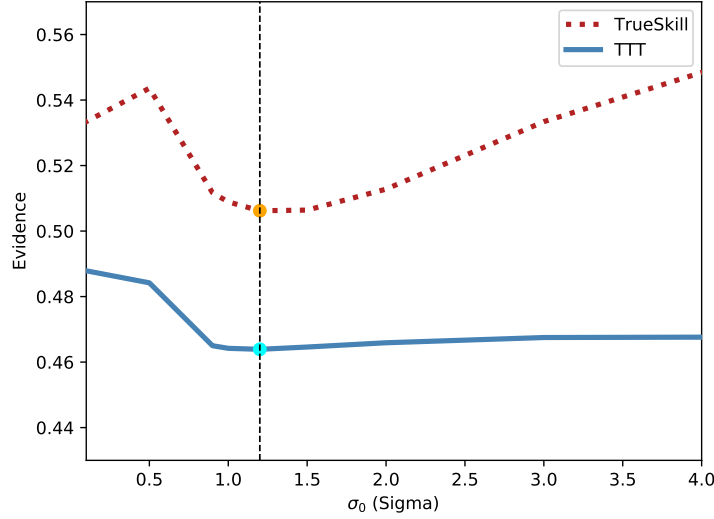


Figura 3.5: Evidencia en función del σ_0 . En cada valor de la curva se iteró el parámetro γ para encontrar el valor ideal. La línea vertical muestra el valor de σ_0 donde la evidencia es mínima para el método TTT.

Se observa como en este caso, el cambio de evidencia para desviaciones estándares iniciales grandes es muy leve. Algo de gran importancia es el aumento de la sensibilidad en el parámetro γ mientras más chico es el σ_0 . En la figura 3.6 se observa la evolución de este parámetro σ en el agente de estudio a través de sus partidas.

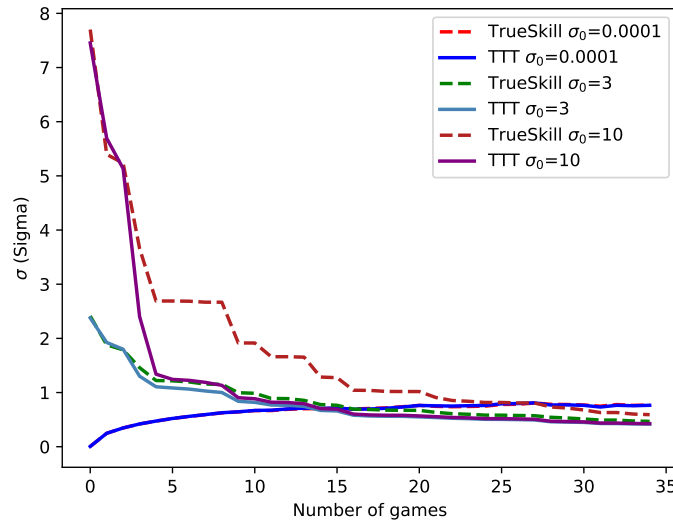


Figura 3.6: Se varía el valor inicial de σ (i.e. σ_0), y se observa cómo responde este parámetro del agente de estudio para distintas configuraciones. Se muestran tres valores de σ_0 iniciales distintos y en cada caso se muestran las curvas para el método TrueSkill y TTT.

Se puede observar que a pesar de inicializarse en valores muy diferentes, logran converger a un valor muy próximo. Una de las grandes diferencias es la cantidad de partidas necesarias para lograr converger al valor de convergencia. Esto muestra la importancia de elegir parámetros de inicialización con un cierto criterio. En nuestro caso usaremos valores mayores al que optimiza la evidencia. Esto se debe a que cada base de datos puede variar estos valores óptimos, y por como vimos, el modelo TTT es más sensible a valores menores del óptimo que a mayores. La figura 3.7 presenta las curvas de aprendizaje para este jugador.

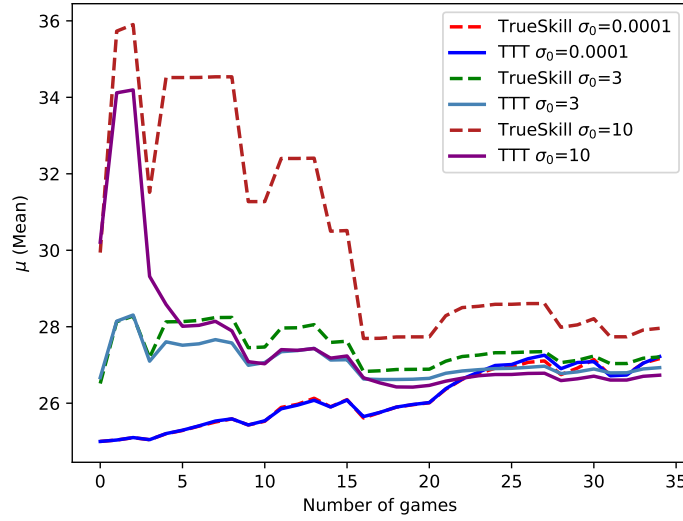


Figura 3.7: Se varía el valor inicial de σ (i.e. σ_0) y se observa cómo responde la media μ del agente de estudio para distintas configuraciones. Se muestran tres valores de σ_0 iniciales distintos y en cada caso se muestran las curvas para el método TrueSkill y TTT.

De nuevo, se puede observar que las curvas logran converger a un mismo valor, exceptuando el caso de TrueSkill con el mayor sigma, que requiere de mayor cantidad de información (partidas) para lograrlo. Se observa la diferente cantidad de partidas necesarias para lograr converger al valor de convergencia en cada caso. Para un mejor entendimiento, se observa en la tabla 3.3, donde se muestra la probabilidad de ganar en cada caso entre dos agentes de estudio.

σ	Evidence TS	Evidence TTT	γ	Probability of winning	Truth probability	$\mu_1 - \mu_2$
0.005	0.53	0.49	3e-01	0.50	0.76	$2.01 \approx 2\beta$
0.010	0.53	0.49	3e-01	0.50	0.76	$2.01 \approx 2\beta$
0.500	0.54	0.48	1e-06	0.64	0.76	$0.53 \approx 0.5\beta$
0.900	0.51	0.46	1e-06	0.69	0.76	$0.76 \approx 0.76\beta$
1.000	0.51	0.46	1e-07	0.70	0.76	$0.80 \approx 0.8\beta$
1.200	0.51	0.46	1e-07	0.71	0.76	$0.85 \approx 0.85\beta$
1.500	0.51	0.46	1e-07	0.72	0.76	$0.90 \approx 0.9\beta$
2.000	0.51	0.47	1e-07	0.73	0.76	$0.95 \approx 0.95\beta$
3.000	0.53	0.47	1e-07	0.74	0.76	$0.99 \approx 0.99\beta$
5.000	0.58	0.48	1e-10	0.75	0.76	$1.03 \approx 1.03\beta$

Cuadro 3.3: Tabla de valores para los parámetros del modelo para diferentes casos de σ_0 diferentes. En estos casos el valor de β es de uno y μ_0 es 25. Se muestra la probabilidad de ganar entre los dos agentes de estudio, la probabilidad real de la simulación de ganar y la diferencia entre medias finales de ambos jugadores y su relación con el parámetro β .

Se puede observar cómo la probabilidad de ganar entre estos dos jugadores varía en relación a la desviación estándar inicial, σ_0 y cómo se relaciona con la probabilidad real de la simulación. A partir del valor de σ_0 con menor evidencia ($\sigma_0 = 0,9$), la diferencia de probabilidad con la realidad disminuye. A pesar de que dicho valor de probabilidad de victoria no presenta el valor más cercano al real, esto no es un problema debido a que este es un único ejemplo de todas las estimaciones hechas en esta simulación. En promedio, cuando consideramos todos los eventos con todos los agentes, el mejor resultado se obtiene con la menor evidencia. Lo que se enfatiza es cómo este parámetro, dentro de un rango, es poco sensible en el modelo.

Como ya hemos comentado, el β es la desviación estándar con la que se computa el rendimiento de los jugadores. Una diferencia de habilidad de un β entre dos jugadores significa que el de mayor habilidad tiene un 76 % de probabilidad de ganar (en el caso en el que la incerteza de los jugadores σ_i sea nula). Intuitivamente esto nos muestra que el cambio del β solo genera un cambio en la escala relativa, manteniendo las mismas probabilidades. Si variamos el β , pero mantenemos la misma relación con σ_0 (de esta forma, omitimos el impacto que tiene este último parámetro), podemos observar el cambio de escala relativa. En la figura 3.8, se observan diversas curvas de aprendizaje para un agente de estudio para el caso de diferentes β , manteniendo una relación de uno sobre seis con σ_0 .

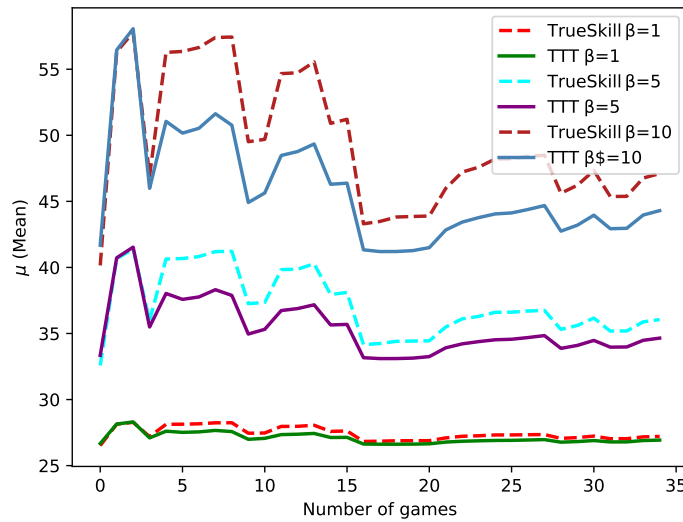


Figura 3.8: Se varía el valor del β , manteniendo una misma relación con el σ_0 y se observa como responde la media del agente de estudio, μ para distintas configuraciones. Se muestran tres valores de β distintos y en cada caso se muestran las curvas para el método **TrueSkill** y **TTT**.

Para poder ver que realmente solo lo que cambió es la escala relativa, se presenta la tabla 3.4.

β	σ_0	Evidencie TS	Evidence TTT	$\mu_1 - \mu_2$	Probability of winning	Truth probaility
0.1	0.3	0.47	0.47	$0.10 \approx \beta$	0.74	0.76
1.0	3.0	0.47	0.47	$1.00 \approx \beta$	0.74	0.76
5.0	15.0	0.47	0.47	$4.98 \approx \beta$	0.74	0.76
10.0	30.0	0.47	0.47	$9.95 \approx \beta$	0.74	0.76
100.0	300.0	0.47	0.47	$99.54 \approx \beta$	0.74	0.76

Cuadro 3.4: Tabla de valores para los parámetros del modelo para diferentes casos de betas diferentes, manteniendo constante la relación con el sigma inicial, siendo $\beta = 3\sigma_0$. En estos casos el valor de γ es de 0,0001 y la media inicial, μ_0 es 25. Se muestra la probabilidad de ganar entre los dos agentes de estudio, la probabilidad real de la simulación de ganar y la diferencia entre medias finales de ambos jugadores, $\mu_1 - \mu_2$, y su relación con el parámetro β .

Se puede ver que, en el caso de una misma relación entre el desviación estándar inicial (σ_0 y β), la evidencia en ambos métodos no varía. La diferencia de habilidad entre estos dos jugadores de estudio, $\mu_1 - \mu_2$, mantienen siempre la misma relación en términos de β y, por consiguiente, la misma probabilidad de ganar. Mostramos así que el único cambio es el de la escala relativa. En este caso, estos dos jugadores fueron simulados con una diferencia exacta de un β de habilidad.

Capítulo 4

Resultados

4.1 Data set

Existen diversos servidores en línea que permiten jugar entre personas de todo el mundo al go, para varios de ellos existe la posibilidad de acceder a la base de datos con la información histórica de las partidas jugadas. Para los propósitos de esta tesis, el servicio en línea a utilizar debía cumplir con ciertas características. Por un lado, requerimos una base de datos, que no solo cuente con un importante número de partidas, sino también que incluya una proporción significativa de partidas en las cuales se haya asignado *handicap*. Además, nos interesa contar con partidas que sean jugadas entre participantes humanos, es así que debemos descartar aquellas protagonizadas por los programas conocidos como *bots*.

Uno de los servidores que cumple con estos requisitos es **KGS** (<http://www.gokgs.com>). En este sitio web se encuentran las partidas almacenadas en diversos archivos comprimidos. Para poder accederlas, se creó un programa del tipo rastreador web (i.e. *crawler*), el cual descargó de manera automática los diferentes archivos.

KGS solo permitió descargar las partidas a partir del año 2007 en las cuales los participantes involucrados poseen rangos de *Dan*. A su vez, estas partidas cuentan con la restricción de que en cada evento uno de los participantes tiene un nivel de *ranking* igual o mayor al de 4to *Dan*. Esta cota inferior temporal de las partidas (i.e. después del 2007) más la restricción en los rangos de los jugadores, implica que la primera partida obtenida por cada jugador no sea la primera partida jugada por éste, en el servidor KGS. Dicho servidor brinda servicio desde el año 2000.

El servicio permite al usuario organizar partidas de dos formas diferentes. La primera, y más sencilla, es cuando dos usuarios generan una partida entre ellos. Como esto no siempre es posible, existe una segunda forma que se basa en el sistema propio de KGS para organizar partidas de forma automática. Esta es la forma más fácil y rápida de encontrar oponentes en KGS, especialmente cuando no se tienen conocidos. Dentro de las opciones de configuración del perfil del usuario se encuentran las preferencias de la búsqueda, como puede ser el tipo de oponente (i.e. humano o *bot*), y la diferencia de rango. Estas propiedades son raramente modificadas por los usuarios, quienes optan por los valores *default*. En promedio, las partidas tienen una diferencia de rango inferior a dos, que coincide con el valor por *default* de configuración de las preferencias de los usuarios. El sistema de *matching* de KGS presenta una lista de partidas disponibles filtradas por las opciones seleccionadas por el usuario y, a partir de esto, el jugador elige en qué partida

participar.

Cada partida está almacenada en un archivo de formato *SGF* (*Smart Game Format*). Este tipo de archivos incluye la información de una única partida codificada por medio de dos letras en mayúscula que representan el tipo de información seguido por el valor de esa propiedad encerrado entre corchetes. Por ejemplo, `SZ[19]` corresponde al tamaño del tablero (las letras `SZ` vienen de la palabra en inglés *size*) y el número indica que el tablero tiene una dimensión de 19×19 casilleros.

El set de datos descargado cuenta con alrededor de dos millones de estos archivos, uno por partida. Para poder analizarlos de manera eficiente, se acumularon los archivos de partida individual en una menor cantidad de archivos, cada uno con 10 000 partidas. Esto fue necesario para poder analizar la información en un tiempo razonable ya que, en caso contrario, por la cantidad de archivos los tiempos de cómputo aumentaban considerablemente. Una vez realizado este procesamiento, se creó un *script* de *Python* que guarda la información relevante de cada partida en una base de datos almacenada en un único archivo *CSV*. Un ejemplo se puede ver en la tabla 4.1.

id	black	white	outcome	black_win	handicap	komi	width	started	whiteRank	blackRank
1472858	MsYoshiko	asaw	W+Resign	0	0.0	0.5	19	2007-01-01	3d	6d
1472859	kiri	asaw	W+Resign	0	0.0	6.5	19	2007-01-01	3d	6d
1472860	Andmore	asaw	B+Resign	1	0.0	6.5	19	2007-01-02	3d	6d
1472861	asaw	Andmore	W+0.50	0	0.0	6.5	19	2007-01-03	2d	5d
1472862	gkang	asaw	B+Resign	1	0.0	6.5	19	2007-01-03	1d	6d
1472863	feez	Rocker	B+4.50	1	2.0	0.5	19	2007-01-03	5d	6d
1472864	feez	Andmore	W+30.50	0	0.0	0.5	19	2007-01-04	2d	6d
1472865	feez	seven11	W+Time	0	2.0	0.5	19	2007-01-05	2d	6d

Cuadro 4.1: Ejemplo de información de partidas obtenidas desde el servicio en línea KGS.

Cada partida está identificada con un número (i.e. un *id*) asignado por el servidor. Luego, incluye el nombre del jugador con ficha negra (*black*) y el nombre del jugador blanco (*white*). El resultado de la partida se muestra con la primer letra del color del ganador, siendo la inicial del color de ficha en inglés, seguido por la forma en la cual la partida finalizó. En el juego go existen tres formas posibles de finalizar una partida:

- i) **Finalización por tiempo:** es cuando la partida supera un determinado valor temporal.
- ii) **Renuncia o abandono:** Se puede finalizar cuando uno de los dos jugadores deja de participar. En la mayoría de los casos se debe a cuando éste deduce que no existe posibilidad de ganar dicha partida.
- iii) **Puntos:** Ambos jugadores acuerdan que no hay jugada posible para mejorar la puntuación del juego en un momento dado. En ese caso, se finaliza la partida y se cuentan los puntos por territorio y capturas para determinar al ganador. Esta es la causa más habitual, ya que es esta puntuación la que condiciona, en la mayoría de casos, al abandono de la partida.

Dentro del archivo procesado, se agregó una columna denominada `black_win` que tiene el valor 1 cuando el jugador de fichas negras es el ganador y 0 en caso contrario. Además, se cuenta con el valor de *handicap* y *komi* utilizados, así como el tamaño del tablero en donde se jugó la partida, la fecha en la cual se inició y el rango de cada jugador.

Una vez obtenida esta base de datos, se pudieron aplicar diversos filtros para generar los subconjuntos de datos utilizados en los diferentes experimentos. En particular, se

eliminaron todas las partidas donde uno de los jugadores era un programa (i.e. un *bot*). Esto representó una disminución de un 15 % del total de la base de datos. Además, se curó la base de datos filtrando casos poco comunes como *handicap* negativo, faltante de datos en alguna partida, *komi* negativo u otras anomalías. Estos totalizaron un número insignificante de partidas filtradas.

■ 4.1.1 Ranking go

El servicio en línea *KGS* utiliza el ranking japonés estándar. Éste se aplica solo para las denominadas partidas clasificadas (*ranked games*), en donde solo juegan jugadores registrados (no se admiten invitados). Además, se utiliza solo el tablero de 19×19 y se agrega un máximo de seis piedras de *handicap*. Se dividen las habilidades en tres categorías, *Kyu*, *Dan* y *Pro Dan*. Las primeras dos son las estándares en el sistema de go, en las cuales un jugador nuevo inicia con el nivel 30 *Kyu*, disminuyendo el valor a medida que mejora hasta llegar al 1er *Kyu*. Luego de éste, la siguiente mejora es 1er *Dan* y, a medida que mejora, este valor sube hasta un máximo de 9no *Dan*. Hasta acá, se considera al jugador como *amateur*. La categoría *Pro Dan* es una escala completamente diferente. Solo se otorga por los organismos de go profesional, estando fuera del sistema de clasificación de los servidores de juego en línea. La cantidad de partidas en nuestra base de datos con jugadores con el rango profesional es tan insignificante que no son de interés.

La diferencia de rango es la que otorga la noción de *handicap* que se requiere para establecer una partida pareja. Por cada rango de diferencia se sugiere una piedra de *handicap*. Se recuerda que el uso de *handicap* tiene como función equiparar la partida en caso de ser necesario, asignando piedras extra en el tablero al principio de la partida, afectando así el juego desde el primer momento. Esto lo difiere de la regla de *komi*, la cual en general tiene como función evitar el empate y anular la ventaja que tiene el jugador negro por inicializar la partida. Esta regla consiste en agregar un número de puntos al finalizar la partida para el conteo de puntos. Estos no tienen ninguna influencia en el juego hasta que esta haya terminado y se cuenten los puntos totales.

Como ya se mencionó, en esta base de datos, en todas las partidas uno de los jugadores tiene un rango mínimo de 4to *Dan*. Es decir, se encuentran partidas con jugadores de menor rango, pero siempre y cuando éste esté jugando en contra de un 4to *Dan* o superior. En la figura 4.1 se puede observar la frecuencia de rango para la primer partida de cada jugador. Como se puede observar, el pico en la cantidad de partidas se da en jugadores de 4to *Dan*, tal como es previsible debido a las características de la base de partidas a la que se puede acceder.

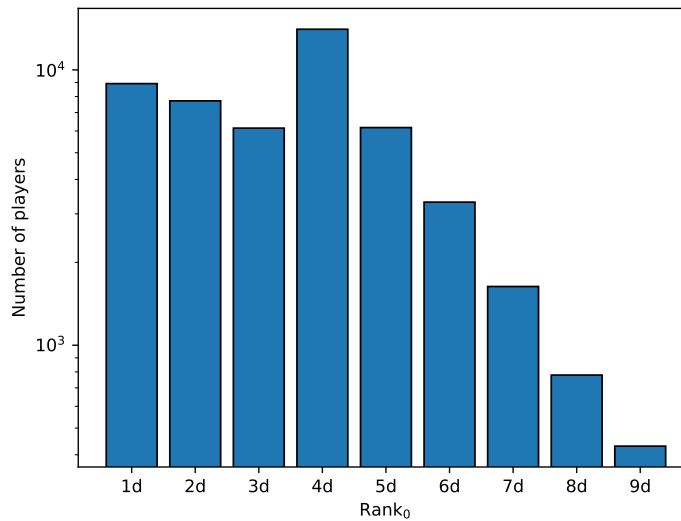
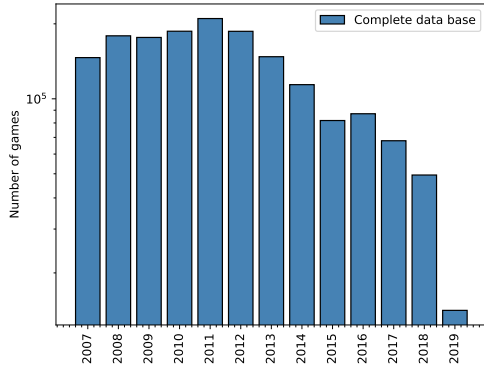


Figura 4.1: Frecuencia de rangos de jugadores en el servicio en línea KGS correspondiente al comienzo del periodo estudiado en este trabajo (1 de enero de 2007).

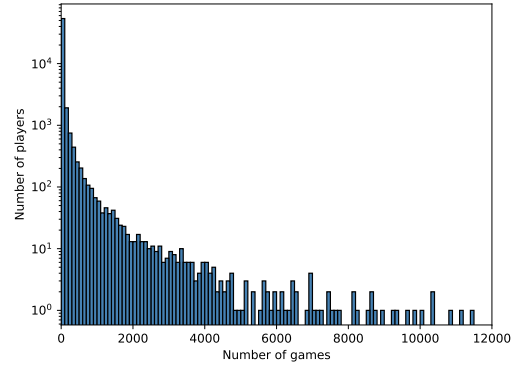
Se observa la distribución centrada en el 4to *Dan* y presenta una asimetría hacia los extremos. El centro de la distribución se debe a que corresponde con el mínimo rango que se necesita para alguno de los dos jugadores en una partida. Un elemento que también debe considerarse para evaluar esta distribución es el significado del rango *Dan*. No cualquier jugador consigue el rango de *Dan*, la mayoría nunca llega a conseguirlo. Se requiere como mínimo un año de juego, pero con este plazo lo consiguen únicamente los más habilidosos. Los que lo logran, en promedio, tardan varios años, incluso algunos más de una década. Esta dificultad también se encuentra para ascender dentro de la categoría de *Dan* y, como es de esperar, hay menos jugadores en las categorías superiores. Esta es la razón por la cual baja considerablemente la frecuencia de jugadores de alto rango en la figura 4.1.

4.1.2 Caracterización

Una vez obtenida nuestra base de datos a utilizar, se realizó su caracterización. Contiene casi dos millones de partidas de go para el tablero estándar de 19×19 . En la figura 4.2(a) podemos ver la cantidad de partidas por año. Se observan partidas desde el 2007 hasta el 2019 teniendo una menor cantidad de partidas en los últimos años. La gran diferencia del número de partidas del último año se debe a que solo se encontraban en el servidor las partidas hasta el mes de abril de dicho año y no el resto de los meses. En la figura 4.2(b) se muestra la cantidad de jugadores con una dada cantidad de partidas. Como es de esperar, se observa mayor cantidad de jugadores con pocas partidas disminuyendo rápidamente.



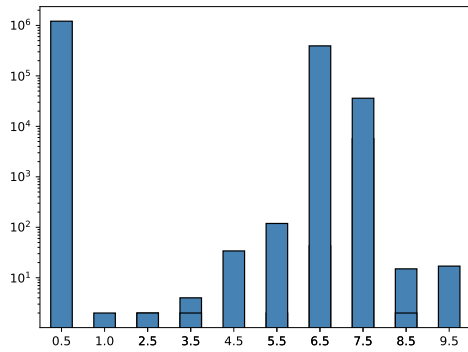
(a) Cantidad total de partidas por año.



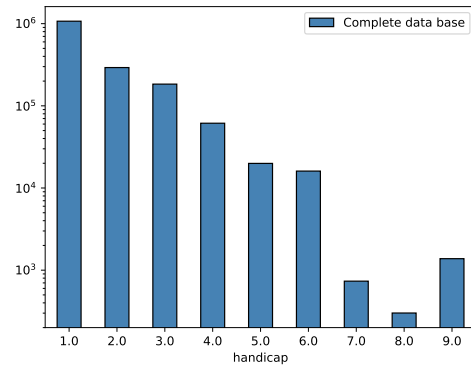
(b) Cantidad de partidas por jugador.

Figura 4.2: Descripción de la base de datos KGS: (a) muestra la cantidad de partidas registradas por año y (b) incluye los jugadores con determinada cantidad de partidas almacenadas en la base de datos.

Algo importante es la frecuencia de los diferentes parámetros del juego. La figura 4.3(a) presenta un histograma de los diferentes *Komis* usados. Se observa, tal como se esperaba, una gran cantidad de partidas para los valores de *Komi* tradicionalmente utilizados (i.e. 0,5, 6,5 y 7,5). Además, en la figura 4.3(b) podemos ver un histograma de los diferentes *handicaps* usados. Se puede ver que los valores de *handicap* más utilizados son los menores, siendo mucho menos frecuente la utilización de *handicap* superiores. A su vez, hay un pico en el valor 9, que corresponde al máximo que se puede utilizar. El valor de *handicap* igual a uno significa que el jugador inicio la partida y no que se le asignó una ventaja extra.



(a) *Komi* asignado a partidas.



(b) *Handicap* asignado a partidas.

Figura 4.3: Descripción de la base de datos KGS:(a) muestra la asignación de *komi* en las partidas descargadas mientras que en (b) se muestra el *handicap*. Es posible que en una misma partida se asigne *komi* y *handicap*.

Del total de partidas analizadas, un 35 % usaron *handicap* mayor a uno, en tanto que un 73 % tienen *komi* igual a 0,5. Por último podemos observar la cantidad de partidas terminadas de las tres formas posibles, esto se observa en la figura 4.4.

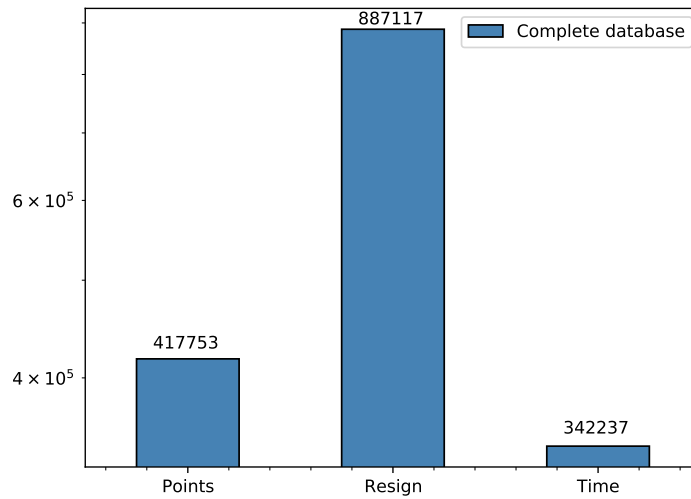


Figura 4.4: Descripción de la base de datos KGS: formas de finalización de las partidas descargadas.

Observamos que la gran mayoría de partidas finalizan con el abandono de uno de los jugadores. Esto se puede deber a que los jugadores de esta base de datos ya poseen un cierto nivel de experiencia, lo cual facilita la lectura del estado de evolución de la partida y, a partir de esto, poder determinar si hay forma de revertir el resultado, o si no le queda otra opción que resignarse y aceptar su derrota.

4.1.3 Curvas de aprendizaje

Por el momento se mencionó la importancia de la habilidad, pero no se profundizó en la evolución que tiene ésta a lo largo del tiempo. La habilidad es adquirida, principalmente, por experiencia individual (*La práctica hace al maestro*). La información con que contamos de un agente es, solamente, la provista por la base de datos de KGS. No conocemos si el agente tiene experiencia previa en otros servicios en línea o si es un ávido jugador de go en plazas públicas o con sus amigos. De todas formas, la intuición básica indica que un jugador nuevo solo puede mejorar y, de esa forma, su curva de aprendizaje inicia con una pendiente positiva. Luego, puede ocurrir que su ritmo de aprendizaje cambie por diferentes motivos, esto se vería como cambios en la pendiente de su curva de aprendizaje.

Entonces, la curva de aprendizaje representa la evolución de la habilidad de un jugador de manera gráfica. Se representa la relación que hay entre el rendimiento de alguien en una tarea, deporte o juego, y la cantidad de experiencia que adquirió. Existen diferentes modelos para su descripción. Uno de los más usados y, a su vez de los más simples, es la que describe una función sigmoide como la que se muestra en la figura 4.5.

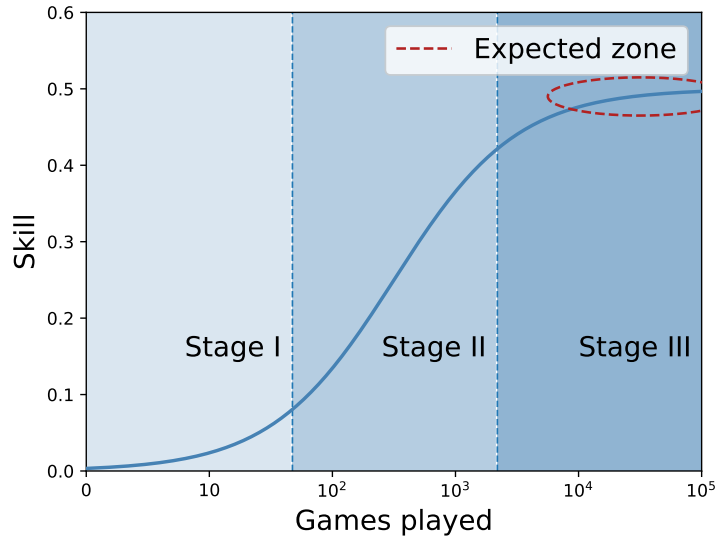


Figura 4.5: Idealización de la evolución del cambio en la habilidad de un jugador respecto a su experiencia. La zona marcada de la curva corresponde con la estimación de la etapa en la que se encuentran los jugadores incluidos en la base de datos bajo estudio. Se resaltan las tres etapas de una curva de aprendizaje sigmoide.

Una curva de aprendizaje normal [23] tiene tres etapas bien definidas: un primer estadio en el cual la persona experimenta (*Stage I*), pero su habilidad no mejora notablemente. En cierto momento, representado aproximadamente por la partida 80 en la figura, el jugador comienza una segunda etapa de mejora en su habilidad caracterizado por un fuerte cambio en su habilidad (*Stage II*). Finalmente, al llegar a la partida 1200, el jugador alcanza su máximo nivel y, a partir de aquí, de existir mejoras resultan marginales (*Stage III*). Se remarca en rojo, la zona de la curva en donde creemos que se encuentran los jugadores de nuestra base de datos. Esto es debido a que tenemos solamente jugadores expertos, en promedio con un *ranking* mayor al del cuarto *Dan*.

4.2 Análisis Handicap

Como se contó en secciones anteriores, el juego go tiene formas de equilibrar las partidas, para que la experiencia de juego sea lo más parejo y desafiante posible. La forma más habitual para hacer este equilibrio consiste en asignar el color de fichas negras al jugador de menor habilidad, el cual inicializa la partida. De esta forma, el jugador negro tiene una ventaja por estar un movimiento por delante del jugador blanco. La regla de **komi**, asigna un puntaje constante que se suma al finalizar la partida en el conteo de puntos con el fin de anular la ventaja por iniciar el juego. A su vez, se suele usar un número no entero para evitar la posibilidad de empate. Un valor de *komi* muy utilizado por jugadores de categoría de *Dan* es el de 0,5 como se observó en la figura 4.3(a). Este valor tiene como objetivo evitar un posible empate sin agregar ventaja adicional. En general dicho valor de *komi* se utiliza cuando existe una diferencia en el rango de los jugadores, haciendo uso de las piedras de ventaja. Éstas son otra forma de equiparar la partida que toma en cuenta la diferencia de habilidad entre jugadores. Dependiendo de que tan grande sea esta diferencia, se agrega piedras de color negro extra al comienzo de la partida, teniendo un impacto

directo en el juego en todo momento. Esto se conoce como **handicap**. El objetivo final de estas reglas es conseguir partidas parejas (i.e. equiprobables) entre ambos jugadores. Esta tesis se enfoca en el estudio y caracterización de las piedras de *handicap*. El modelo de TTT supone a los jugadores con una habilidad representada mediante una distribución, la cual posee una media que representa la verdadera habilidad y una desviación estándar, representando la incertidumbre que se tiene de la habilidad. A su vez, en cada evento, cada agente presenta un rendimiento representado por una distribución de misma media que la habilidad pero con una misma desviación estándar en todo evento y para todo agente. El jugador que posee el mayor rendimiento en un dado evento es el ganador. Dicho modelo no contempla al *handicap* en sus estimaciones y el impacto que producen en las partidas. En este trabajo, se propone agregar esta información al modelo de estimación. El *komi* será objeto de un futuro estudio similar, pero queda fuera del alcance de la presente tesis.

Esta idea de introducir al modelo de estimación la información del *handicap* resulta novedosa debido a que en la comunidad de go utilizan sistemas que no consideran explícitamente el *handicap*. Solo en algunos casos lo incorporan mediante técnicas ad-hoc. Estos modelos suelen considerar que la diferencia de rango entre los jugadores equivale a la cantidad de piedras necesarias para equilibrar la partida. Al agregar al modelo información adicional del *handicap*, se logró caracterizar y definir los valores de habilidad de las piedras de ventaja. Con esto se lograría poder asignar de forma más certera los valores de *handicap* en futuras partidas y así generar partidas más parejas y desafiantes para los jugadores. Otro efecto que ocurre al estar considerando mayor información de cada evento, es una mejora en la evidencia del modelo, logrando estimaciones más certeras.

La propuesta para atacar este problema es agregar al modelo el *handicap* contando a las piedras de ventaja como jugadores que conforman un equipo con quien recibió el *handicap*. Cada cantidad de piedras de ventaja diferente se la considera un jugador distinto. Es decir, dos piedras de *handicap* es un jugador al que se le va a estimar su habilidad, como así también al jugador de tres piedras de *handicap* y así sucesivamente. De esta forma, no se hace ninguna suposición sobre la forma funcional que tendrá el comportamiento de la habilidad de las fichas de ventaja en función de la cantidad utilizada. Otra alternativa hubiera sido considerar a cada piedra de ventaja como un solo jugador independientemente de la cantidad utilizada. Luego, se formaría un equipo entre el jugador negro y la cantidad de jugadores que representan la cantidad de piedras de ventaja usadas a modos de *handicap*. Esta aproximación cuenta con un comportamiento lineal ya que la habilidad del equipo sería la del jugador más la habilidad de una única piedra tantas veces como piedras utilizadas. Debido a esto, se optó por el modelo libre descripto anteriormente en el que a cada grupo de piedras se lo considera un jugador distinto y su habilidad es estimada de forma independiente. De esta forma, como veremos más adelante, este comportamiento lineal surge de forma natural.

Una de las suposiciones que se tuvo en cuenta y se lo agregó al modelo, es que el rendimiento de la piedra de ventaja es siempre la misma a diferencia del rendimiento de los jugadores. Es por ello que se modela al *handicap* con un parámetro β nulo. A su vez, otra suposición agregada al modelo es que éstas tienen un valor de habilidad único sin poder mejorar en el tiempo. En ambos casos, estas suposiciones, mejoran levemente la evidencia. Para ello se modela al *handicap* con un parámetro γ nulo. Una vez hecho el análisis y obtenido el valor de habilidad que tiene una piedra de *handicap* en una dada escala, resulta sencillo asignar la ventaja más justa en un dado evento intentando tener una partida lo más equiprobable posible.

4.2.1 Implementación

Se analizaron los parámetros del modelo, con la condición de tener la mínima evidencia posible para esta base de datos. Se recuerda, tal como se mencionó en la sección 2.1.1 (Modelo probabilístico), que la evidencia (ecuación 2.10) es menos el logaritmo de la productoria de la probabilidad de que suceda el resultado de cada partida de acuerdo a las estimaciones del método. La metodología de evaluación usada es mediante la minimización de la misma. En particular, se estudió la relación de los parámetros β y σ_0 , para determinar en cada caso el parámetro γ que minimiza la evidencia. Debido al costo computacional que tiene el método, se estudió la relación de estos parámetros con una muestra de la base de datos de un 33 %. Además, como se mostró en la sección anterior, los métodos TrueSkill y TTT, tienen un comportamiento similar en relación a los parámetros y la evidencia. Es por ello que se utilizó esta propiedad para realizar un primer estudio utilizando solamente TrueSkill, el cual muestra un menor tiempo de cómputo. Por como está formado TTT incluye al método TrueSkill. En la figura 4.6, se muestran tres relaciones diferentes de los parámetros β y σ_0 utilizando el método TrueSkill. Se normalizaron las curvas con su máximo debido a que la diferencia en escala absoluta dificultaba la visualización de las formas funcionales. En cada caso se resalta el valor de γ que minimiza la evidencia.

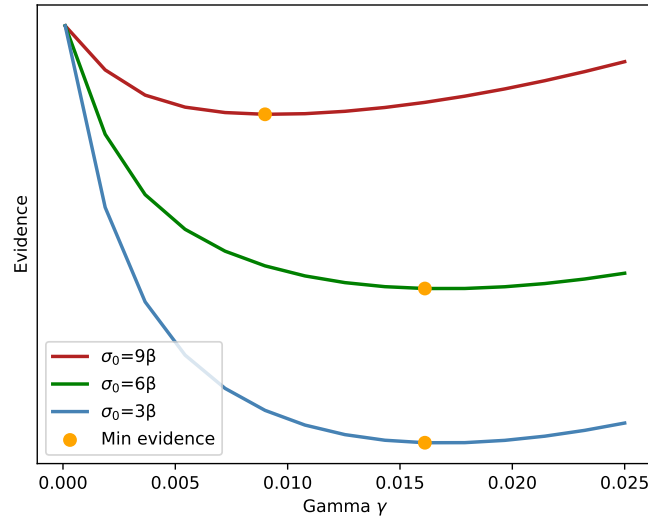


Figura 4.6: Se varía la relación entre el β y el σ_0 y se observa la variación de la evidencia para un barrido del parámetro γ . Se muestran tres relaciones diferentes y en cada caso se muestran las curvas para el método TrueSkill con el valor del parámetro γ que minimiza la evidencia. Las curvas están normalizadas con respecto al máximo valor de cada curva para una mejor visualización.

Se puede observar que a mayor constante multiplicativa entre los parámetros, mayor la evidencia. Utilizando la relación que minimiza la evidencia, se realizó un barrido más fino del parámetro γ cercano al valor hallado. Esta vez se realizó tanto para el método TrueSkill como TTT. En ambos casos, se aplicó el método propuesto en esta tesis, agregando la información del *handicap* y sin ésta. Esto se visualiza en la figura 4.7

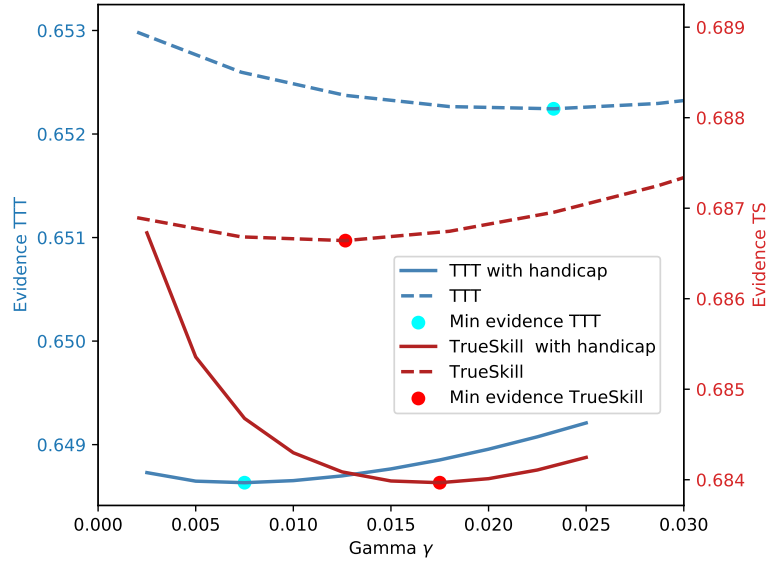


Figura 4.7: Se utiliza la relación $\sigma_0 = 3\beta$ recién analizada y se observa la variación de la evidencia para un barrido del parámetro γ entorno al valor recién hallado provista por el método TrueSkill. Se incluyen los métodos TrueSkill y TTT con y sin *handicap*. Se muestra cada curva de cada método con su respectivo eje y el valor de γ que minimiza la evidencia.

Utilizando una misma relación entre el parámetro σ_0 y β , el valor de γ que minimiza la evidencia, es similar entre las diferentes implementaciones mostradas. A su vez se observa una mejora en evidencia de parte del método TTT de la forma propuesta en esta tesis (i.e. agregando la información del *handicap*). Este último es el que se utiliza en el resto de la tesis utilizando el valor del parámetro $\gamma = 7,5 \times 10^{-3}$ el cual minimiza su curva de evidencia. Con ello se aplicó el método a nuestra base de datos completa, utilizando los parámetros recién estudiados. La incorporación del *handicap* al modelo, permitió estimar y caracterizar el comportamiento de habilidad de las fichas de ventajas, como también una mejora en evidencia al estar considerando mayor información de los eventos. Las estimaciones obtenidas para cada valor de *handicap*, están representadas mediante una distribución gaussiana, al igual que las estimaciones de habilidad de los jugadores. Observando la relación entre las medias de estas distribuciones en función de la cantidad de piedras de ventaja, se observó una relación lineal. Se seleccionaron mil puntos normalmente distribuidos de cada gaussiana, y se ajustó mediante una regresión lineal por mínimos cuadrados. En la figura 4.8, se muestran las estimaciones de habilidad de dicho caso, en partidas de tablero estándar 19×19 .

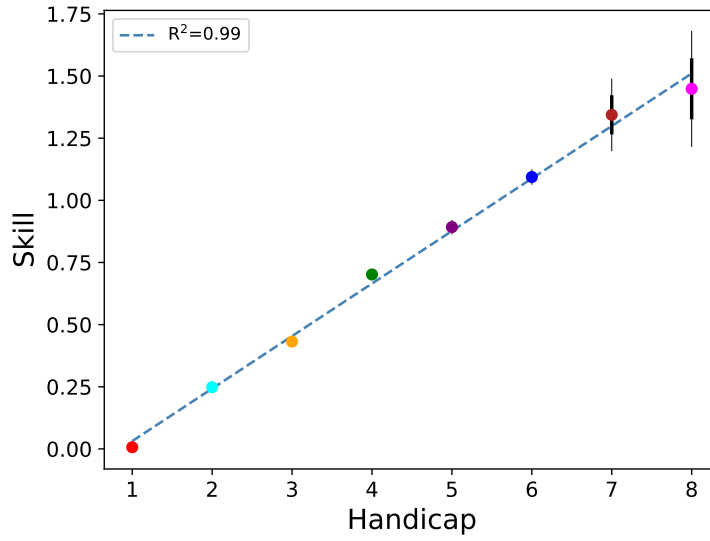


Figura 4.8: Estimación de habilidad de *handicap* utilizando el método TTT propuesto. Se muestra la estimación de habilidad de cada valor de *handicap* junto con su ajuste lineal. Se representa a la media μ de cada distribución con círculos de colores. Las barras de error son la desviación estándar σ y el doble de ésta.

En la figura se puede observar un comportamiento lineal en las distribuciones de habilidad de ventaja. El caso de *handicap* 9 se omitió en el ajuste debido a que éste representa el valor máximo de ventaja posible en el juego, involucrando partidas en las cuales podrían requerir un mayor valor de *handicap*. Por este motivo se obtenían estimaciones poco representativas. Conociendo la habilidad que agrega cada piedra de *handicap*, se puede estimar la ventaja que requiere una partida futura para que el evento sea lo más equiprobable y desafiante posible. Para lograr ésto de una forma más sencilla, se cambió la escala relativa mediante el parámetro β (manteniendo constante su relación con σ_0), obteniendo una nueva relación lineal del *handicap* con pendiente uno. Esto no es otra cosa que utilizar el método con un nuevo parámetro β igual al recién utilizado dividido por la pendiente de la recta de la figura 4.8. Esto se muestra en la figura 4.9

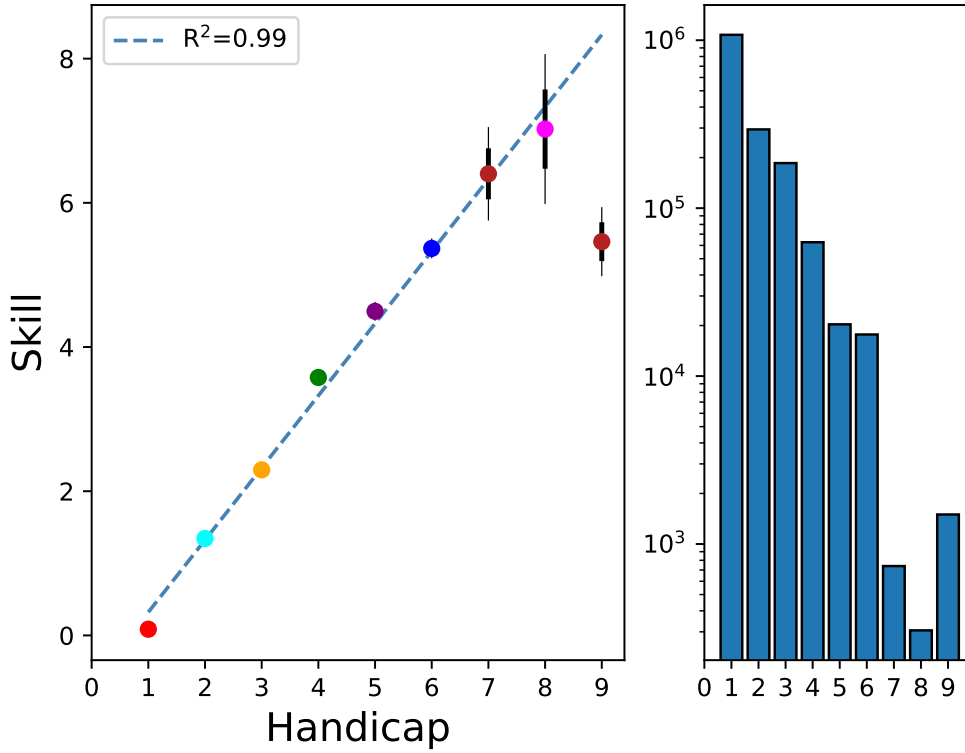


Figura 4.9: Estimación de habilidad de los diferentes handicaps con el método TTT con *handicap*. Se utiliza el parámetro β con información de la pendiente de la figura 4.8. Se muestra la estimación de habilidad de cada valor de *handicap* con su ajuste lineal con pendiente uno. Se representa a la media μ de cada distribución con círculos de colores. Las barras de error son la desviación estándar σ y el doble de ésta. Pendiente $a=0,997 \pm 0,002$ con ordenada al origen $b=-0,560 \pm 0,009$

Se puede observar la relación lineal de *handicap* con pendiente uno. Por el mismo motivo anteriormente explicado, se omitió el valor de ventaja 9 en el ajuste. Con esto llegamos a la conclusión que utilizando un valor de β de 4,525, el valor de habilidad de una piedra de ventaja, sin considerar por inicializar la partida (i.e. *handicap* 1), es de 1,4383. Esto representa el 31,7% del valor de β . A su vez, se estimó que la ventaja en habilidad que tiene el jugador negro por inicializar la partida es de 0,4383 el cual representa el 9,6 % del valor de β .

Utilizando nuestro sistema de habilidad y estimaciones finales, se compara la asignación de *handicap* propuesto y el utilizado en el servidor KGS. Se analiza la diferencia de partidas en las cuales se asignó una diferente cantidad de *handicap* que el estimado y los resultados. El estudio consistió en calcular la probabilidad de victoria del jugador negro de cada evento utilizando todos los posibles valores de habilidad de *handicap*. El cálculo de probabilidad se realizó mediante la ecuación 2.27, la cual considera la diferencia de habilidad e incertezas a priori estimada por el modelo en un dado evento, entre el jugador blanco y el equipo formado por el jugador negro y el valor de *handicap*. Para los análisis futuros en esta tesis, se tomó la probabilidad que surge al utilizar el valor de ventaja utilizado en dicho juego y el valor de *handicap* que produce la probabilidad más cercana al equiprobable, considerando a este último como el valor propuesto. Se compararon las estimaciones de probabilidad en cada evento con la frecuencia de victorias por parte del jugador negro. Para ello se

separaron las partidas en 100 intervalos de probabilidad estimada, y en cada uno se calculó el porcentaje victorias del jugador negro. Esto se visualiza en la figura 4.10.

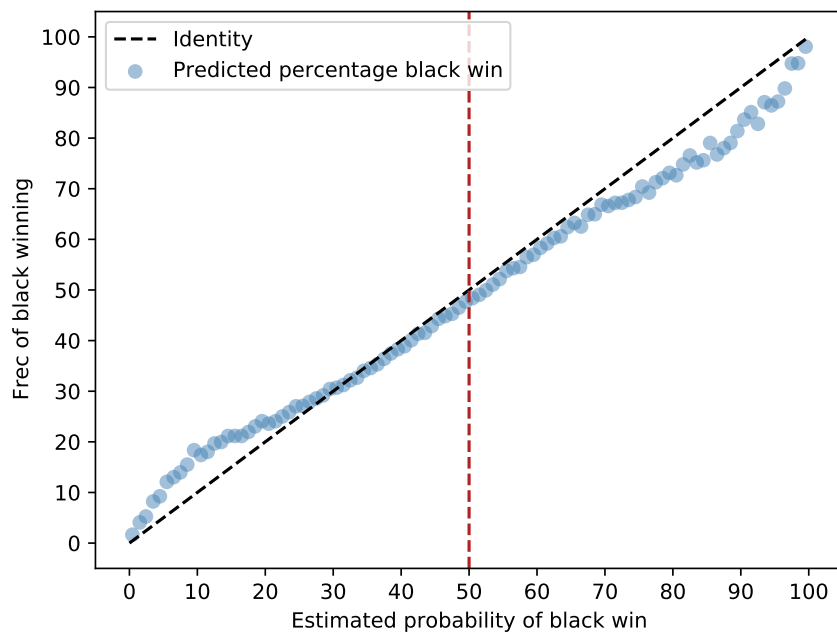


Figura 4.10: Intervalos de probabilidad de victoria estimada del jugador negro mediante el método TTT en relación a la fracción de partidas ganadas por dicho jugador en cada intervalo. Se muestra la recta identidad, mostrando la coincidencia teórica perfecta en el límite de poseer información infinita.

Se puede observar como nuestras predicciones se asemejan a la identidad, teniendo un mayor desvío en las puntas debido a poseer una información acotada, como se mostró en la sección validación, figura 3.1. Esto genera una confianza del modelo utilizado. Además, se compararon las distribuciones de frecuencia de eventos con una dada probabilidad de ganar utilizando la ventaja propuesta, la utilizada por KGS y la frecuencias de victorias del jugador negro en cada intervalo de KGS. Esto se visualiza en la figura 4.11.

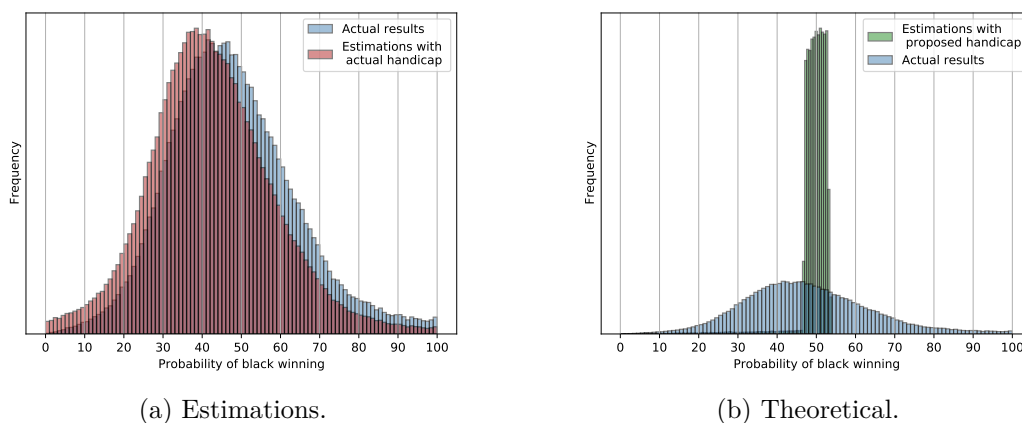


Figura 4.11: Frecuencia de probabilidades normalizadas. En verde se muestra la frecuencia de probabilidades utilizando el valor de *handicap* propuesto. En rojo se observa la frecuencia de probabilidades estimadas utilizando el valor de *handicap* de KGS. En azul se observa la frecuencia de victorias por el jugador negro en cada intervalo de probabilidad estimada de KGS.

Se observa en la figura 4.11(a) cómo la distribución de probabilidades estimada (roja) se asemeja a la distribución de frecuencias de victorias en los eventos de KGS, mostrando un correcto comportamiento a lo esperado. Esta distribución se encuentra centrada en 47,6 %, mientras que la otra se encuentra centrada en 42,1 %, con una desviación estándar similar. Dicha distribución estimada difiere del equiprobable, en favor del jugador blanco (i.e. por debajo del 50 %), con una desviación estándar menor al 10 %. Por otro lado, en la figura 4.11(b), se muestra en verde la distribución de probabilidades estimadas de los eventos utilizando el *handicap* propuesto. En este caso, la media se encuentra centrada en 49,7 %, con una desviación estándar menor al 2,5 %. Con esto, se puede determinar que, utilizando el *handicap* propuesto, se obtiene una distribución cuya media se encuentra cercana al valor equiprobable. A su vez esta distribución cuenta con una menor incerteza, lo que implica una mayor frecuencia de partidas en condiciones de equiprobabilidad. Mostrando un gran aumento en la frecuencia de partidas con probabilidades cercanas a la deseada. Esta distribución es de carácter teórico, debido a que estas partidas ya fueron jugadas, sin posibilidad de observar los resultados utilizando el valor de *handicap* propuesto.

En nuestra base de datos tenemos un 35 % de partidas con las que se jugaron con *handicap*. Según los valores de *handicap* propuesto, el total de partidas que requerían ventaja es del 67,5 %. Es decir, nuestra predicción suponen una mayor cantidad de partidas con la utilización de *handicap*. De las partidas que se jugaron con *handicap*, el modelo propuesto coincide en un 89,6 % de las partidas. Esto significa que aproximadamente nueve de cada diez partidas que KGS asignó *handicap*, el modelo propuesto también lo hizo. Para el caso de las partidas en las cuales se estima que el jugador negro requería el uso de *handicap* y no se le fue asignado, se observa que solo el 36 % de éstas fueron ganadas por el jugador negro. Esto indica que dichas partidas deberían haber hecho uso del *handicap* al igual de lo que predice el modelo propuesto.

Se realizaron diversos estudios filtrando por el tipo de *handicap* que se utilizó en los eventos y el propuesto. Luego se visualiza por cada valor de ventaja el porcentaje de victorias que tuvo el jugador negro. En todos los casos se visualiza en el eje x el valor de *handicap* propuesto. Para el primer caso, se filtraron por partidas que se jugaron sin ventaja (i.e. *handicap* = 1). En la figura 4.12 podemos observar la frecuencia de victorias

de parte del jugador negro.

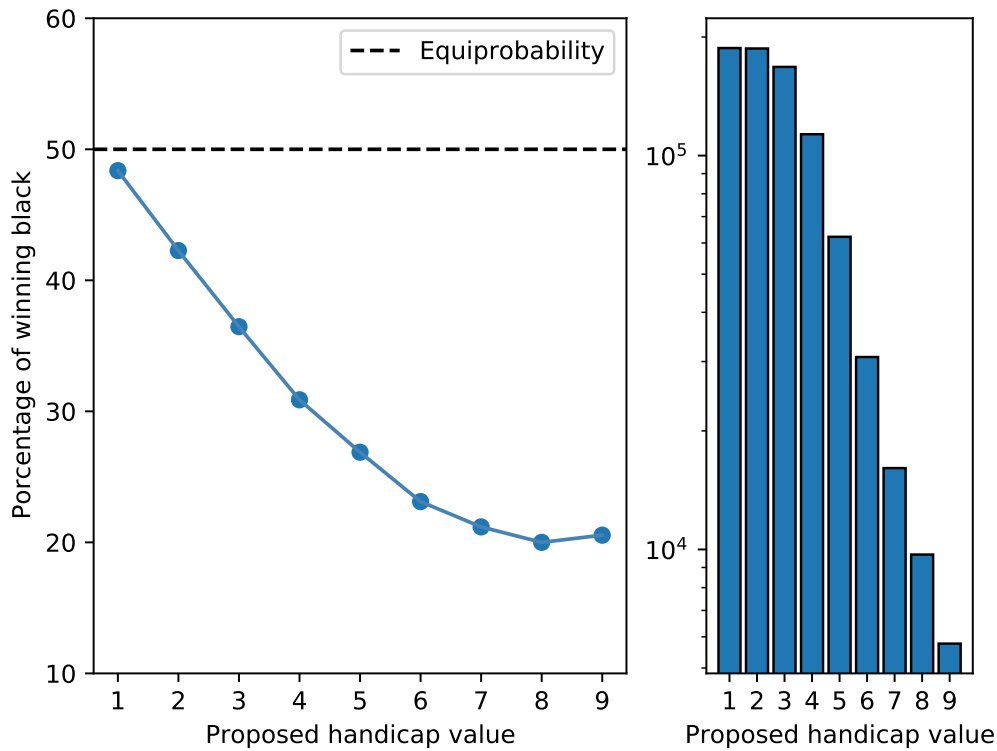


Figura 4.12: Partidas jugadas con *handicap* uno. Se muestra el porcentaje de victorias del jugador negro en base al valor de *handicap* que nuestro modelo hubiera asignado.

Se puede observar cómo en las partidas que se estima que no necesitaban *handicap* y además fueron jugadas sin *handicap*, tienen un porcentaje de victorias cercanas al equiprobable. En cambio, a medida que se hubiera requerido un mayor valor de *handicap*, el porcentaje de victorias obtenidas por el jugador negro disminuye notablemente, llegando a probabilidades cercanas a 20 % para un *handicap* estimado mayor al de 8. La curva de porcentaje de victorias de negro muestra una forma amesetada que se vuelve notoria a partir de 7. Este resultado es consistente con lo esperado, ya que a mayor valor de ventaja estimado existe una mayor diferencia de habilidad entre los jugadores, implicando menor probabilidad de victoria para el jugador negro.

Con esta misma idea, se aplicó otro filtro a la base de datos, en este caso analizando las partidas que recibieron el mismo valor de *handicap* que lo que hubieran recibido de acuerdo a nuestra estimación. La figura 4.13 muestra que los porcentajes de victoria están cercanos al equiprobable.

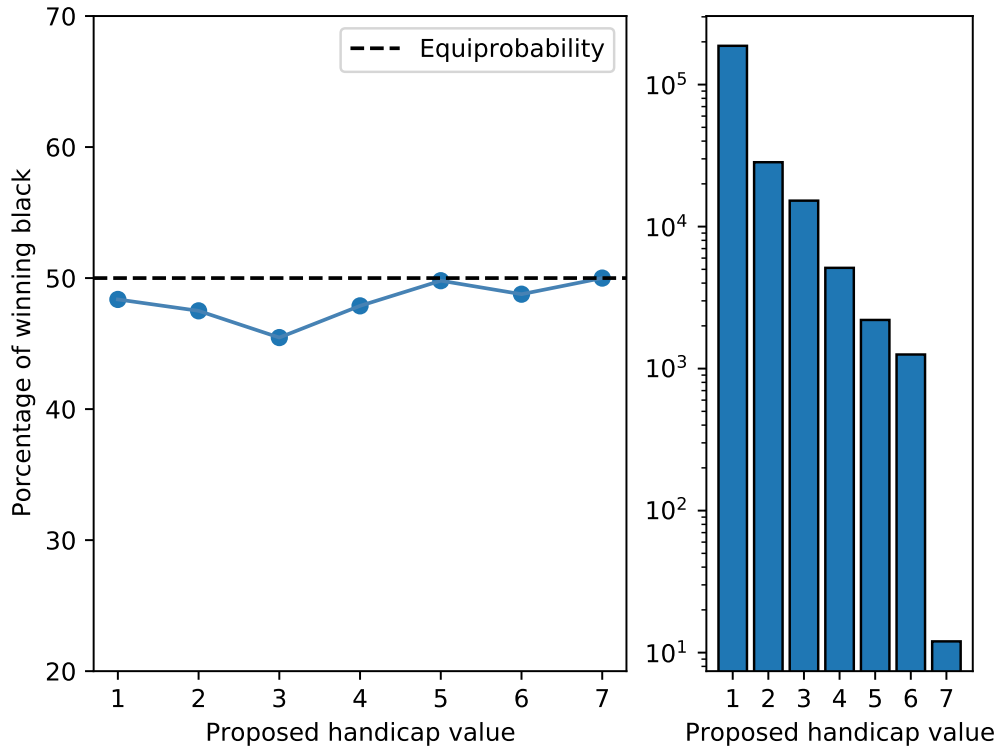


Figura 4.13: Porcentaje de victoria del jugador negro en función al handicap propuesto, en partidas que el handicap utilizado coincidía con este.

Algo que resalta es el valor de 45 % en el caso de *handicap* 3. Una posible explicación para este fenómeno se ensayará en el siguiente análisis. Luego de esto, se filtraron las partidas en las cuales KGS asignó una cantidad de *handicap* mayor o menor a lo propuesto por nuestro modelo, calculando la frecuencia de victorias en cada caso. En la figura 4.14 se puede visualizar el primer caso en el cual el jugador negro recibió mayor *handicap* que lo propuesto por nuestro modelo. Se muestran los casos en que esta sobreasignación es por una única piedra o por más de una.

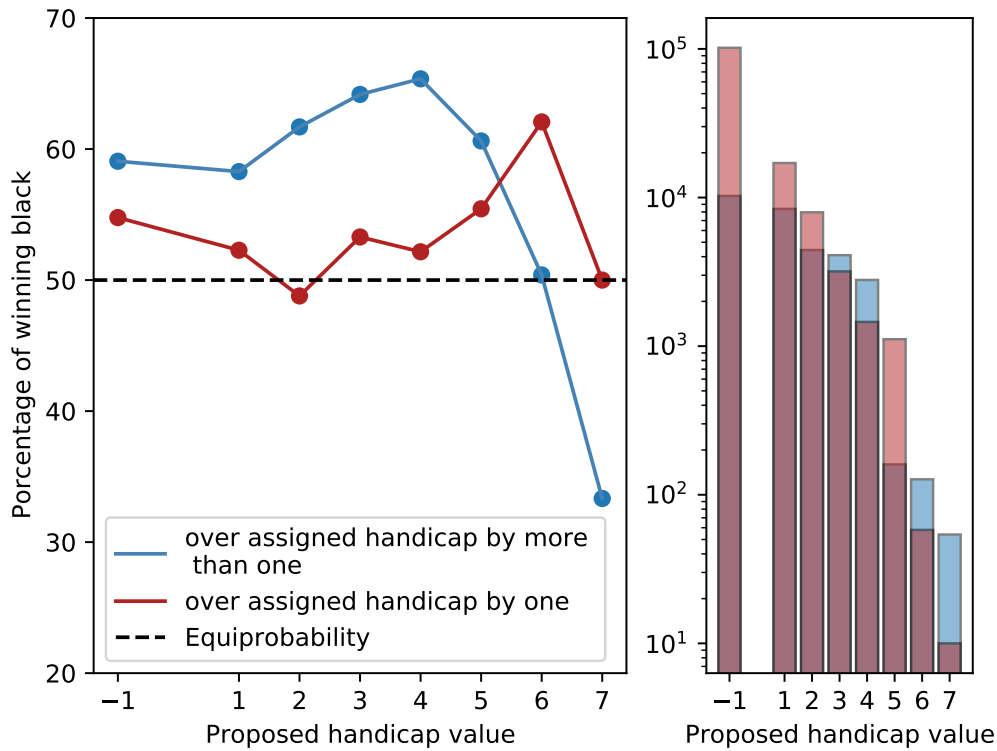


Figura 4.14: Partidas en la cual se estima que se utilizó un *handicap* mayor al necesario. Porcentaje de victoria del jugador negro en función al *handicap* propuesto. En rojo se muestra la curva donde se asignó una única piedra de más y en azul una cantidad mayor.

Debido a que el sistema asignó mayor *handicap* de lo necesario, estas partidas presentan una ventaja a favor del jugador negro. En concordancia con esto, en la figura 4.14 se observa un porcentaje de victoria por arriba del equiprobable a favor de este jugador. Se muestran, a su vez, los casos en los que el *handicap* propuesto y el utilizado difieren en una única piedra, observándose también un porcentaje de victorias superior al equiprobable pero inferior al caso anterior. Las anomalías encontradas en los valores de *handicap* 6 y 7 en ambas curvas se las atribuye a la poca cantidad de partidas incluidas. A su vez, se presenta una probabilidad menor al equiprobable para el caso de *handicap* 2 propuesto (*handicap* 3 asignado). Un análisis complementario de esto último podría venir de considerar la información del *komi* de los eventos, pero este estudio queda fuera del alcance de la presente tesis. En la figura 4.15 se puede visualizar el segundo caso en el cual el jugador negro recibió un menor valor de *handicap* al propuesto, como también los casos en los cuales solamente recibió una única piedra de *handicap* de menos.

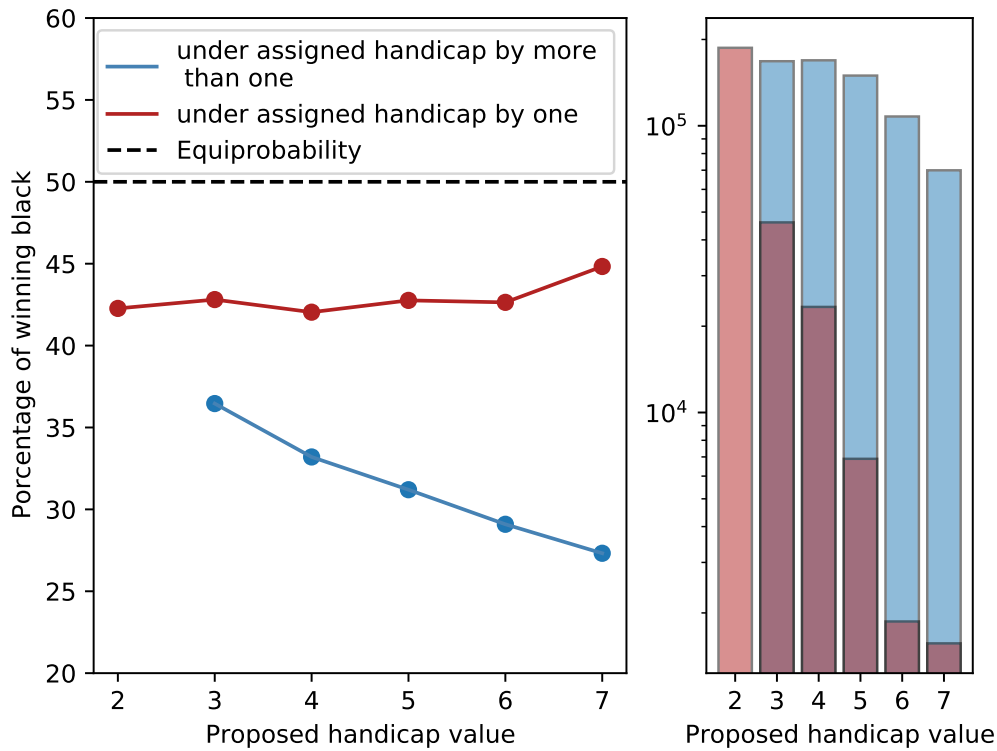


Figura 4.15: Partidas en la cual se estima que se utilizó un *handicap* menor al necesario. Porcentaje de victoria del jugador negro en función al *handicap* propuesto. En rojo se muestra la curva donde se asignó una única piedra de menos y en azul una cantidad menor aún.

Debido a que el sistema asignó un menor *handicap* de lo necesario, estas partidas estuvieron a favor del jugador blanco, mostrando en todos los casos un porcentaje de victoria por debajo del equiprobable a favor de este jugador. Como se puede observar, a diferencia del caso de sobreasignación de *handicap*, los porcentajes de victoria tienen una mayor estabilidad. Esto indica que la diferencia de una piedra de *handicap* resulta en una diferencia de probabilidad de victoria aproximada de 7,17%.

En la curva azul se observa una pendiente negativa, mostrando que a mayor valor de *handicap* propuesto menor la frecuencia de victorias del jugador negro. Dicha curva involucra partidas en las cuales se subasignó el valor de *handicap* con una diferencia mayor a una piedra que el valor propuesto. Ésto implica que a mayor valor de *handicap* propuesto ésta diferencia pueda ser por más cantidad de pierdas. Es decir, a mayor valor de *handicap* propuesto, mayor es la diferencia de habilidad posible entre los jugadores. Ésto incluye a partidas con una menor probabilidad de victoria para el jugador negro, mostrando una menor frecuencia tal como se puede observar en la figura 4.15.

Por último, se analizaron las partidas en las cuales se estima una incorrecta asignación de color de piedras, es decir, el jugador negro poseía una mayor habilidad que el jugador blanco. Para ello, en la estimación de *handicap* propuesto, se permitieron valores negativos, calculados a partir de la recta de la figura 4.9, asignando ventaja al jugador blanco. En la figura 4.16 se pueden visualizar los eventos en los cuales la asignación de fichas debió ser contraria a lo realizado y, a su vez, partidas que hubiesen requerido ventaja.

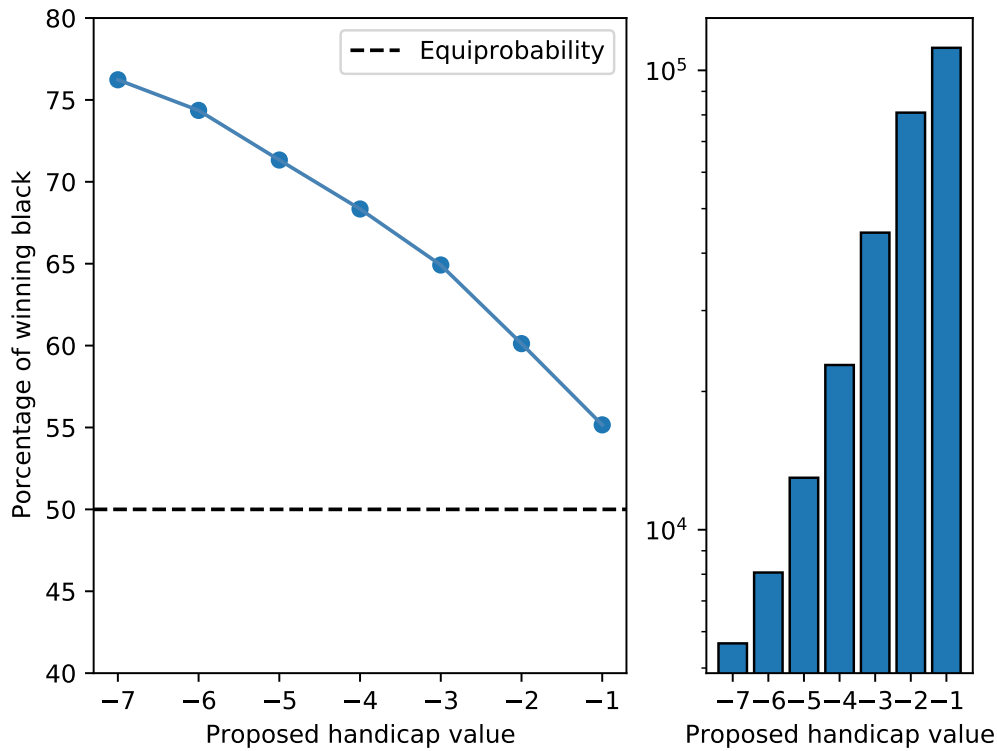


Figura 4.16: Partidas en la cual se estima que la asignación de color de piedras fue incorrecto. Porcentaje de victoria del jugador negro en función al *handicap* propuesto. Se muestra que a mayor estimación de ventaja para el jugador blanco, mayor la frecuencia de victorias del jugador negro.

Se puede observar cómo, en todos los casos, el porcentaje de victorias del jugador negro supera el equiprobable. A mayor valor de *handicap* propuesto mayor el porcentaje de victorias del jugador negro. En la gran mayoría de este conjunto de partidas, el sistema de KGS solo asignó de forma incorrecta el color sin agregar ventaja adicional (i.e. *handicap* = 1).

Como se pudo mostrar, el sistema de ventajas propuesto, resulta consistente con los datos del servidor en línea KGS. Se pudo observar que las probabilidades estimadas en los eventos, resulta consistente con las frecuencias de victorias por parte del jugador negro. A su vez, se mostró que la distribución de probabilidades teórica utilizando el valor de *handicap* propuesto se encuentra centrado en el equiprobable con una desviación estándar considerablemente menor a la estimada recién comentada. Con ello, se realizaron diversos filtros entre el *handicap* propuesto y utilizado, mostrando el comportamiento de la frecuencia de victorias en cada uno de estos. Se logró mostrar concordancia en cada uno de estos de acuerdo a lo esperado.

4.2.2 Distribución inicial y final

Se comparó la distribución de rangos iniciales de los jugadores de nuestra base de datos con las estimaciones provistas por TTT propuesto en esta tesis. Para ello, se tomaron las estimaciones *priors* de la primer partida de cada jugador. Debido a la propagación de información de este método, estos *priors* logran representar la habilidad de cada jugador en esta primer partida debido a la convergencia del modelo. Luego para poder tener una analogía con los nueve rangos de *Dan*, se dividió a la distribución estimada en nueve intervalos equiespaciados. Esto se puede observar en la figura 4.17.

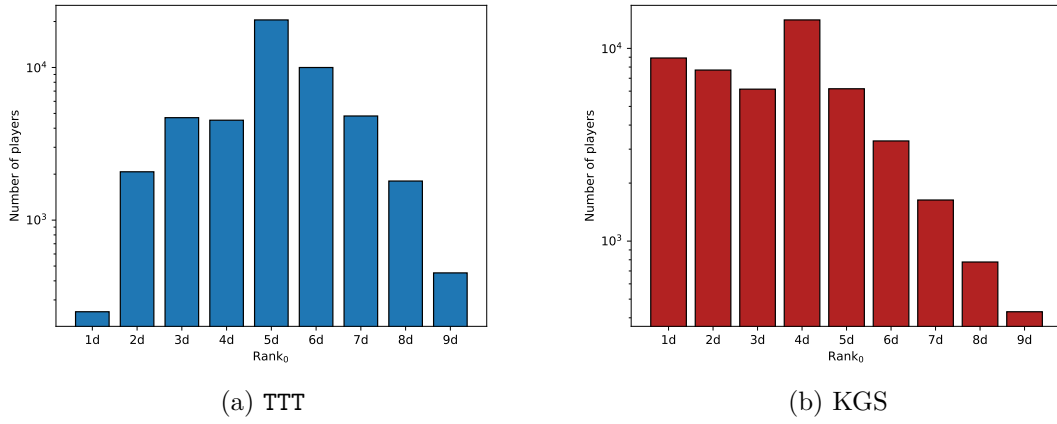


Figura 4.17: Distribución de habilidades iniciales de la población de KGS. En (a) se muestran las estimaciones realizadas con TTT y en (b) se observa la distribución de los rangos del propio servidor.

Estas dos distribuciones difieren en su media, la cual en el caso estimado se encuentra en un rango más alto que en el de KGS. Se observa una mayor simetría del tipo de distribución campana en el caso de la propuesta. Utilizando la misma analogía se pueden visualizar las distribuciones finales de habilidad y de rango de los jugadores de la base de datos. Es decir, se observa la última partida jugada por cada jugador. Esto se puede observar en la figura 4.18.

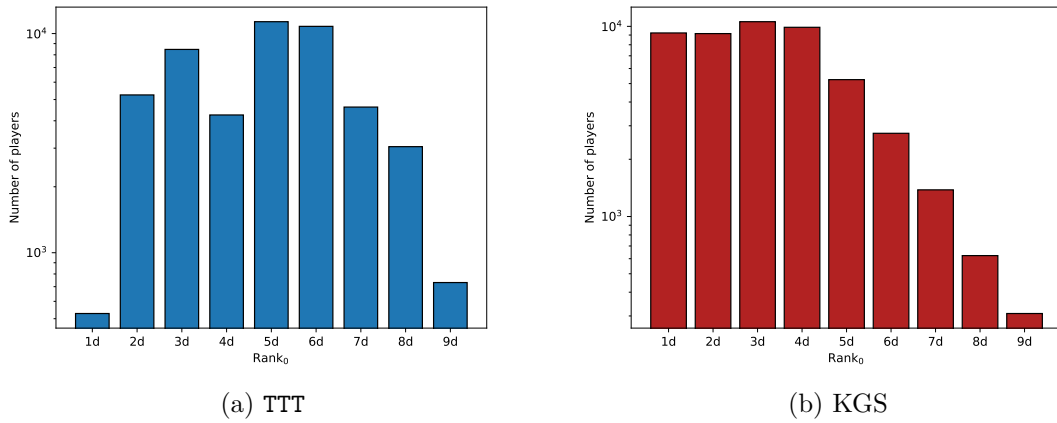


Figura 4.18: Distribución de habilidades iniciales de la población de KGS. En (a) se muestra las estimaciones realizadas con TTT y en (b) se observa la distribución de los rangos del propio servidor.

En el caso de la distribución estimada, ésta se asemeja a la inicial con una varianza más grande. Esto se condice con las curvas de aprendizaje descripta anteriormente, en donde los jugadores de la base de datos se encuentran en la etapa final del aprendizaje, y se estima poco cambio de habilidad y de rango de los jugadores, mostrando así distribuciones similares. En relación a la distribución de KGS, esta presenta una reducción en el 4to *Dan*, con una mayor cantidad de jugadores en rangos menores a este. Un comportamiento similar se produce con respecto al 5to *Dan* de las estimaciones por *TTT*.

■ 4.2.3 Análisis predictivo

Para poder probar el carácter predictivo del modelo propuesto se simuló su uso de manera *online*. Se seleccionó una cantidad de partidas utilizando diversos criterios y se utilizó el resto de la base de datos para estimar la habilidad de los jugadores involucrados y así calcular la probabilidad de victoria en cada evento. Se realizaron dos estudios diferentes con dos conjuntos de características distintas para la selección de partidas. En ambos casos se seleccionaron las últimas partidas jugadas de la base de datos en las cuales participen jugadores distintos. Debido a que la condición de jugadores diferentes es muy estricta, filtrando una gran cantidad de partidas, se permitió repetir los jugadores una única vez. La condición para esto es que el jugador repetido tenga color de ficha diferente en los dos eventos seleccionados. A su vez se seleccionaron jugadores en los cuales, en dicho evento, el conocimiento del modelo sobre éste sea grande, es decir un σ_i relativamente chico ($\sigma_i \leq 1$). Debido a que el método *TTT* propaga la información al pasado, la noción de conocimiento de un jugador en un dado evento, no es representativo. Esto se debe a que un jugador con muchas partidas va a poseer un σ_i chico aun en su primer partida. Es por eso que la selección de este parámetro es utilizando *TrueSkill*. Se recuerda que por como esta formado *TTT* incluye al método *TrueSkill*, permitiendo guardar esta información. La diferencia principal de los estudios consistió con la diferencia de rango entre los jugadores en cada evento. En ambos casos se seleccionaron un total de 500 partidas. En el primer caso se seleccionaron partidas en las cuales los participantes tenían diferencia de rango de KGS. El segundo caso se seleccionaron partidas en las cuales los participantes tenían el mismo rango y sin la utilización de *handicap*.

Las partidas seleccionadas se guardaron en un archivo separado, las cuales se las denominó partidas de estudio. Debido a que el modelo *TTT* propaga la información temporalmente, se optó por remover todas las partidas en las cuales participe alguno de los jugadores seleccionados en partidas posteriores a la escogida. De esa forma se asegura que el modelo no tenga información sobre los jugadores de estudio en partidas posteriores a la que se quiere estimar, simulando de mejor forma las predicciones *online*. La base de datos sin estas partidas ni las partidas seleccionadas se las denominó de entrenamiento. Con ello se implementó el método *TTT* propuesto estimando los últimos *posteriors* de los jugadores

escogidos. Se utilizaron éstas estimaciones como *priors* de las partidas de la base de estudio y en cada una se calculó la probabilidad de victoria del jugador negro, considerando el valor de *handicap* utilizado. Estas probabilidades se separaron en intervalos, calculando la frecuencia de victorias del jugador negro en cada una de éstos. Dada la cantidad de partidas seleccionadas, se optó por intervalos de 10 %. Los resultados del primer escenario, donde los participantes tienen rangos diferente en cada evento, se observa en la figura 4.19,

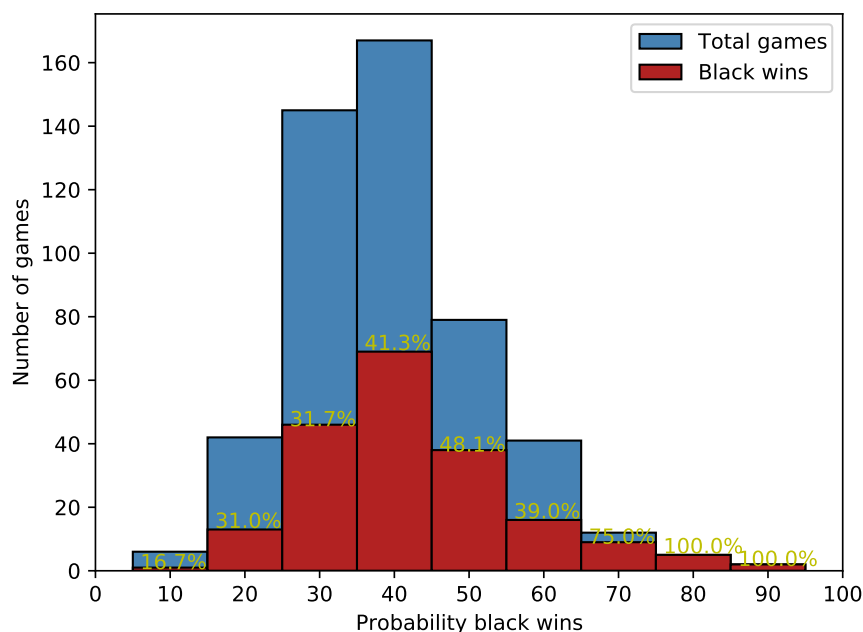


Figura 4.19: Frecuencia de probabilidades junto con la frecuencia de victoria del jugador negro, en partidas en la cual el rango del servicio en línea KGS de los jugadores era diferente. Se muestra el porcentaje de victorias en cada intervalo.

Se puede observar una distribución centrada en 40 %. Se muestra en azul el número de partidas en un dado intervalo. En rojo se muestra el número de partidas ganadas por el jugador negro con su respectivo porcentaje de victorias en cada intervalo. En cada intervalo, se pueden observar valores cercanos al esperado. El intervalo centrado en 60 %, presenta una diferencia considerable al valor esperado. No se encontró explicación aparente a este fenómeno. En la figura 4.20 se observa el segundo escenario.

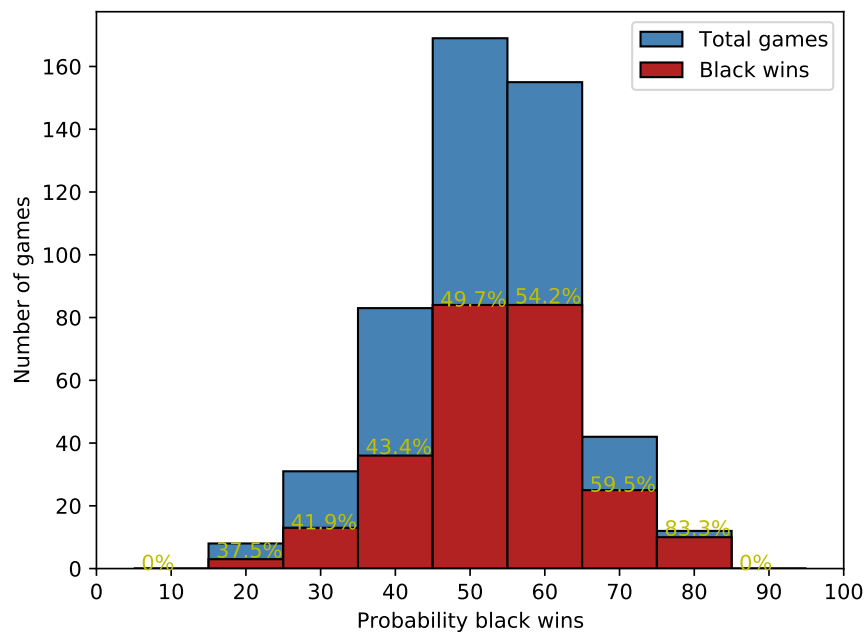


Figura 4.20: Frecuencia de probabilidades junto con la frecuencia de victoria del jugador negro, en partidas en la cual el rango del servicio en línea KGS de los jugadores eran iguales. Se muestra el porcentaje de victorias en cada intervalo.

Se observa una distribución centrada en el equiprobable. Nuevamente se presenta una concordancia entre los intervalos y la frecuencia de victorias. De esta forma se logra validar el carácter predictivo del modelo. En trabajos futuros se pretende realizar un análisis similar considerando el valor de *komi*, pudiendo de esta forma captar la totalidad de información disponible en cada evento logrando estimaciones más certeras y por lo tanto mejores predicciones.

Capítulo 5

Conclusiones y trabajo a futuro

La tesis presentada propuso un modelo basado en **TrueSkill Through Time (TTT)** con el fin de analizar la asignación de *handicap* en el juego de go en base a información de partidas recolectadas de un servicio en línea. Modelando los posibles valores de *handicap* como jugadores adicionales que hacen equipo con el jugador más débil, se logró estimar no sólo las habilidades de los jugadores involucrados si no a su vez, el valor de habilidad que aportan las fichas de ventaja. Esto permitió definir una nueva forma de asignación de *handicap* en las partidas de go. Al comparar nuestra propuesta de asignación de *handicap* con la tradicional, se mostró que con nuestro enfoque se pueden lograr partidas más parejas.

En el capítulo 2 se mostró la teoría necesaria para entender el método bayesiano **TrueSkill** y su sucesor **TrueSkill Through Time**. Se mostraron algunas de las propiedades básicas de las distribuciones gaussianas, como también los conceptos utilizados en el método como los gráficos de factores, el pasajes de mensajes entre sus nodos a través del algoritmo *sum-product*, la metodología de evaluación del modelo, entre otros. Debido a la poca bibliografía de este último, se enfatizó la descripción de las diferentes variables y parámetros de **TrueSkill**, como también el funcionamiento conceptual y analítico del método, que resulta fundamental para el correcto entendimiento de su predecesor. En particular, se buscó un enfoque diferente a la bibliografía encontrada con el objetivo de poder ser utilizado por futuros estudiantes interesados en el tema. Se finalizó con una visualización de la actualización de los *priors* para un ejemplo genérico.

En el capítulo 3 se validó tanto el modelo como la implementación utilizada. Este capítulo fue de suma importancia debido a que el software utilizado, desarrollado por el laboratorio LICAR del Departamento de Computación, se encuentra en una etapa de testeo. Para ello se realizaron diversas simulaciones, logrando comparar de forma satisfactoria las estimaciones halladas con las simuladas. Se mostró que con la métrica de evaluación elegida asociada con la minimización de la evidencia del modelo, se obtienen estimaciones correctas. Por otra parte, se estudiaron los impactos y efectos que tienen los diferentes parámetros en las distribuciones de habilidad actualizadas como también la relaciones entre éstos. Se mostró como el parámetro μ inicial representa un sesgo en la escala absoluta de las medias de habilidad, como también el efecto que tiene el parámetro γ en la evidencia. Con esto se encontraron diferentes relaciones de importancia, las cuales se utilizaron en los análisis posteriores teniendo un criterio a la hora de elegir los parámetros a utilizar.

El capítulo 4 se separó en dos partes. En la primera se caracterizó la base de datos usada y también se analizó el sistema de ranking utilizado por el servidor y la comunidad de go. En particular, se observó que los jugadores involucrados en esta tesis, son expertos

en el juego según el sistema de ranking del servidor (i.e. rango *Dan*). Esto muestra que estos jugadores se encuentran posiblemente al final de la curva de aprendizaje, implicando variaciones chicas de habilidad en el tiempo. En la segunda parte se detalló la propuesta propia de esta tesis, en la cual se agrega la información del *handicap* al modelo. Ésta involucra considerar a los valores de *handicap* como jugadores, los cuales forman parte del equipo del jugador más débil, estimando de esta forma ambas habilidades. Debido a que en el juego go al jugador más débil se le asigna el color de fichas negro y los jugadores *handicap* formar equipo con éste, enfrentándose al jugador blanco. Se observó una disminución en la evidencia, en relación al método sin considerar al *handicap* como un jugador, mostrando así estimaciones más certeras según nuestra metodología de evaluación. Utilizando las relaciones estudiadas en el capítulo 3 se realizó un breve análisis sobre los parámetros que minimizan la evidencia en esta base de datos en particular, siendo éstos los utilizados en los análisis finales. Aplicando éstos, se estimaron las habilidades de los diferentes valores de *handicap*, observándose un comportamiento lineal. Teniendo las habilidades estimadas de los jugadores más la de los *handicap* provista por la caracterización lineal, se calculó la probabilidad de victoria del evento para los posibles valores de ventaja buscando el valor que más se acerque al equiprobable. El valor encontrado para cada evento es el que se considera como el valor de ventaja propuesto por nuestro modelo. Se encontró que en el 59 % de los casos este valor se diferencia al utilizado por el servidor KGS. A través de diferentes estudios se logró comprobar la consistencia del modelo propuesto.

Uno de estos estudios consistió en calcular la probabilidad de victoria en cada evento de uno de los jugadores, en particular el de ficha negra. Estos eventos se separaron en intervalos de acuerdo a su probabilidad calculada y en cada uno de éstos se calculó la frecuencia de victorias de las fichas negras. Al haber obtenido valores similares se logró visualizar la eficacia del modelo. Otro estudio consistió en realizar diferentes filtros, agrupando los eventos entre el *handicap* utilizado en cada evento y el propuesto, y se calculó la frecuencia de victorias del jugador negro según el *handicap* propuesto. Ésto mostró en cada caso ser consistente con lo esperado. En los casos en los cuales el modelo propuesto consideró un mayor valor de *handicap* al usado, se encontró una frecuencia de victorias por debajo al equiprobable, y en el caso de un menor valor de *handicap*, se encontró que esta frecuencia se encontraba por arriba.

Para lograr simular el uso del método propuesto de forma *online*, se realizó un análisis predictivo. Utilizando y separando un conjunto de partidas a las que se las denominó de estudio, se estimó el resto de la base de datos denominada de entrenamiento. De esta manera se obtuvieron las habilidades de esta base de datos filtrada y se utilizaron las últimas estimaciones de cada jugador de estudio como *priors* para las partidas de estudio. Se calcularon las probabilidades de victorias de estas partidas, separándolas en intervalos y se calcularon las fracciones de victorias en cada uno de estos. Se encontró que las probabilidades estimadas y las fracciones de victorias eran similares, mostrando un correcto comportamiento.

En este análisis, se encontraron algunas anomalías no esperadas en determinados casos. Entre éstas hubieron algunas diferencias en la probabilidad calculada mediante el método propuesto y la frecuencia de victorias. Se propuso como trabajo futuro a esta tesis la incorporación de la información del *komi* utilizado en los eventos como también el impacto del tamaño temporal de los lotes. La implementación de éstas no se encuentran estudiadas. Una posibilidad para el primer caso es realizar un estudio análogo, agregando además al *komi* como un jugador formando equipo con el jugador blanco. El efecto del *komi* no se produce desde el principio de la partida como lo es en el caso del *handicap*. Para el segundo caso, se variarían los tamaños de los lotes buscando minimizar la evidencia. El

tamaño del mismo debe aproximar el tiempo promedio en que un agente requiere para obtener un cambio significativo en su habilidad. Debido a que en cada variación, se requiere encontrar el parámetro γ ideal, el costo computacional de dicho estudio es alto, siendo necesario buscar técnicas y formas para reducir los tiempos. Una posibilidad podría ser el uso de la unidad de procesamiento gráfico (*graphics processing unit*) para los cálculos, siendo necesario adaptar el software para ello. Además este estudio abre las puertas para estudiar un comportamiento dinámico del parámetro γ , el cual dependa de la cantidad de lotes entre una partida y otra de un mismo agente. De esta forma, el modelo considera el desconocimiento adquirido por la ausencia del mismo en un dado tiempo. Al considerar una mayor cantidad de información de cada evento (*komi*) y utilizar el lote temporal que mejor aproxima la realidad de los agentes en el cambio de su habilidad, se podrían responder alguna de estas anomalías y a su vez minimizar aun más la evidencia. De esta forma se podrían mejorar las predicciones del modelo tanto de los jugadores como de la habilidad del *handicap*. Así se obtendría un sistema de estimación más certero el cual permitiría lograr generar juegos más desafiantes para los participantes.

Bibliografía

- [1] B. A. Beheim, C. Thigpen, R. McElreath, Strategic social learning and the population dynamics of human behavior: The game of go, *Evol Hum Behav* 35 (5) (2014) 351–357. [doi:10.1016/j.evolhumbehav.2014.04.001](https://doi.org/10.1016/j.evolhumbehav.2014.04.001).
- [2] M. Derex, R. Boyd, The foundations of the human cultural niche, *Nat Comm* 6 (1) (2015) 8398. [doi:10.1038/ncomms9398](https://doi.org/10.1038/ncomms9398).
- [3] J.-F. Gariépy, K. K. Watson, E. Du, D. L. Xie, J. Erb, D. Amasino, M. L. Platt, Social learning in humans and other animals, *Front Neurosci* 8 (58). [doi:10.3389/fnins.2014.00058](https://doi.org/10.3389/fnins.2014.00058).
- [4] L. Glowacki, L. Molleman, Subsistence styles shape human social learning strategies, *Nat Hum Behav* 1 (5) (2017) 0098. [doi:10.1038/s41562-017-0098](https://doi.org/10.1038/s41562-017-0098).
- [5] P. J. Richerson, R. Boyd, J. Henrich, Gene-culture coevolution in the age of genomics, *Proc Natl Acad Sci USA* 107 (Supplement 2) (2010) 8985–8992. [doi:10.1073/pnas.0914631107](https://doi.org/10.1073/pnas.0914631107).
- [6] R. Boyd, P. J. Richerson, J. Henrich, The cultural niche: Why social learning is essential for human adaptation, *Proc Natl Acad Sci USA* 108 (Supplement 2) (2011) 10918–10925. [doi:10.1073/pnas.1100290108](https://doi.org/10.1073/pnas.1100290108).
- [7] M. E. Glickman, Parameter estimation in large dynamic paired comparison experiments, *J Roy Stat Soc C-App* 48 (3) (1999) 377–394. [doi:10.1111/1467-9876.00159](https://doi.org/10.1111/1467-9876.00159).
- [8] A. E. Elo, *The Rating of Chess Players, Past and Present*, Ishi Press, 2008.
- [9] R. Herbrich, R. Minka, T. Graepel, Trueskill: A bayesian skill rating system, in: *Proc of the Conf in Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, 2006, pp. 569–576.
- [10] A. E. Elo, *The USCF rating system*, U.S. Chess Federation publication, 1961.
- [11] P. Dangauthier, R. Herbrich, T. Minka, T. Graepel, Trueskill through time: Revisiting the history of chess, in: J. C. Platt, D. Koller, Y. Singer, S. T. Roweis (Eds.), *Proc of the Twenty-First Annual Conf on Neural Information Processing Systems*, Curran Associates, Inc., 2007, pp. 337–344.
- [12] L. L. Thurstone, A law of comparative judgment., *Psychological review* 34 (4) (1927) 273.
- [13] E. Zermelo, The calculation of the results of a tournament as a maximum problem in the calculus of probability (translation of 1929 paper), in: H.-D. Ebbinghaus, A. Kanamori (Eds.), *Ernst Zermelo - Collected Works Volume II*, *Schriftmath*, Springer, [1929] 2013. [doi:10.1007/978-3-540-70856-8](https://doi.org/10.1007/978-3-540-70856-8).

- [14] R. A. Bradley, M. E. Terry, Rank analysis of incomplete block designs: I. the method of paired comparisons, *Biometrika* 39 (3/4) (1952) 324–345. doi:10.2307/2334029.
- [15] F. Mosteller, Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations, *Psychometrika* 16 (1) (1951) 3–9. doi:10.1007/BF02313422.
- [16] M. Glickman, Example of the glicko-2 system, <http://www.glicko.net/glicko/glicko2.pdf>, accessed: 2016-07-12 (11 2013).
- [17] W. J. van der Linden, *Handbook of Item Response Theory, Volume 1: Models, Statistics in the Social and Behavioral Sciences*, Chapman and Hall/CRC, 2016.
- [18] J.-P. Fox, *Bayesian item response modeling: Theory and applications*, Statistics for Social and Behavioral Sciences, Springer, 2010. doi:10.1007/978-1-4419-0742-4.
- [19] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, Berlin, Heidelberg, 2006.
- [20] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, Factor graphs and the sum-product algorithm, *IEEE T Inform Theory* 47 (2) (2001) 498–519. doi:10.1109/18.910572.
- [21] J. Pearl, Fusion, propagation, and structuring in belief networks, *Artif Intell* 29 (3) (1986) 241–288. doi:10.1016/0004-3702(86)90072-X.
- [22] R. Herbrich, On gaussian expectation propagation, Tech. rep., Microsoft Research (Jul. 2005).
- [23] A. Newell, P. S. Rosenbloom, J. R. Anderson, *Cognitive skills and their acquisition*, Psychology Press, 1981, Ch. Mechanisms of skill acquisition and the law of practice, pp. 1–55.

Apéndice A

Propiedades de las gaussianas

A.1 Propiedades

Simetría: Si evaluó una gaussiana en un dado valor p_i , vale que

$$\mathcal{N}(p_i|\mu_i, \beta^2) = \mathcal{N}(\mu_i|p_i, \beta^2) \quad (\text{A.1})$$

De esto mismo se puede pensar con la función acumulativa Φ , haciendo cambio de variable, $k = p - p_i$

$$\begin{aligned} \Phi(p_i|\mu, \sigma^2) &= \int_{-\infty}^{p_i} \mathcal{N}(p|\mu, \sigma^2) dp = \int_{-\infty}^{p_i} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(p-\mu)^2}{\sigma^2}} dp \\ &= \int_{-\infty}^{k=p_i-p_i=0} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(k-(\mu+p_i))^2}{\sigma^2}} dk = \Phi(0|\mu - p_i, \sigma^2) \end{aligned} \quad (\text{A.2})$$

Es por esto que Si tengo la integral de una gaussiana evaluada en un dado valor p_i , vale la siguiente,

$$\int \mathcal{N}(p_i|s_i, \beta^2) ds_i = \int \mathcal{N}(s_i|p_i, \beta^2) ds_i \quad (\text{A.3})$$

A.2 Suma de n normales

Sabemos que

$$\begin{aligned} t_n &= \sum_{i=1}^n x_i \\ \int \cdots \int \mathbb{I}(t_n = \sum_{i=1}^n x_i) \left(\mathcal{N}(t_n|\mu_t, \sigma_t^2) \prod_{i>1}^n \mathcal{N}(x_i|\mu_i, \sigma_i^2) \right) dt_n dx_2 \dots dx_n &= \mathcal{N} \left(x_1 | \mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2 \right) \end{aligned} \quad (\text{A.4})$$

Queremos probar por inducción.

$$P(n) := \int \cdots \int \mathbb{I}(t_n = \sum_{i=1}^n x_i) \left(\prod_{i=1}^n \mathcal{N}(x_i | \mu_i, \sigma_i^2) \right) dx_1 \dots dx_n \stackrel{?}{=} \mathcal{N} \left(x_1 | \mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2 \right) \quad (\text{A.5})$$

Casos base

$$P(1) := \int \mathbb{I}(t_1 = x_1) \mathcal{N}(t_1 | \mu_t, \sigma_t^2) dt_1 = \mathcal{N}(x_1 | \mu_t, \sigma_t^2) \quad (\text{A.6})$$

Luego $P(1)$ es verdadera.

$$\begin{aligned} P(2) &:= \iint \mathbb{I}(t_2 = x_1 + x_2) \mathcal{N}(t_2 | \mu_t, \sigma_t^2) \mathcal{N}(x_2 | \mu_2, \sigma_2^2) dx_1 dx_2 \\ &= \int \mathcal{N}(x_1 + x_2 | \mu_t, \sigma_t^2) \mathcal{N}(x_2 | \mu_2, \sigma_2^2) dx_2 \\ &= \int \mathcal{N}(x_1 | \mu_t - x_2, \sigma_1^2) \mathcal{N}(x_2 | \mu_2, \sigma_2^2) dx_2 \\ &= \int \mathcal{N}(x_2 | \mu_t - x_1, \sigma_1^2) \mathcal{N}(x_2 | \mu_2, \sigma_2^2) dx_2 \\ &\stackrel{*}{=} \int \underbrace{\mathcal{N}(\mu_t - x_1 | \mu_2, \sigma_t^2 + \sigma_2^2)}_{\text{const.}} \underbrace{\mathcal{N}(x_1 | \mu_*, \sigma_*^2)}_1 dx_1 \\ &= \mathcal{N}(-x_1 | \mu_2 - \mu_t, \sigma_t^2 + \sigma_2^2) \\ &= \mathcal{N}(x_1 | \mu_t - \mu_2, \sigma_t^2 + \sigma_2^2) = \mathcal{N}(x_1 | \mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2) \end{aligned} \quad (\text{A.7})$$

Donde $*$ vale por la demostración de multiplicación de normales en la sección Luego, vale $P(2)$.

Paso inductivo $P(n) \Rightarrow P(n+1)$

Sea,

$$\begin{aligned} P(n) &:= \int \cdots \int \mathbb{I}(t_n = \sum_{i=1}^n x_i) \left(\mathcal{N}(t_n | \mu_t, \sigma_t^2) \prod_{i>1}^n \mathcal{N}(x_i | \mu_i, \sigma_i^2) \right) dt_n dx_2 \dots dx_n \\ &= \mathcal{N} \left(x_1 | \mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2 \right) \end{aligned} \quad (\text{A.8})$$

Queremos ver que vale $P(n+1)$

$$\begin{aligned} P(n+1) &= \int \cdots \int \mathbb{I}(t_{n+1} = x_{n+1} + \sum_{i=1}^n x_i) \left(\mathcal{N}(t_{n+1} | \mu_t, \sigma_t^2) \prod_{i=1}^n \mathcal{N}(x_i | \mu_i, \sigma_i^2) \right) \\ &\quad \mathcal{N}(x_{n+1} | \mu_{n+1}, \sigma_{n+1}^2) dt_n dx_2 \dots dx_n dx_{n+1} \end{aligned} \quad (\text{A.9})$$

Por independencia

$$\begin{aligned}
&= \int \mathcal{N}(x_{n+1}|\mu_{n+1}, \sigma_{n+1}^2) \left[\int \cdots \int \mathbb{I}(t_{n+1} = x_{n+1} + \sum_{i=1}^n x_i) \right. \\
&\quad \left. \left(\mathcal{N}(t_{n+1}|\mu_t, \sigma_t^2) \prod_{i=1}^n \mathcal{N}(x_i|\mu_i, \sigma_i^2) \right) \right. \\
&\quad \left. \mathcal{N}(x_{n+1}|\mu_{n+1}, \sigma_{n+1}^2) dt_n dx_2 \dots dx_n \right] dx_{n+1}
\end{aligned} \tag{A.10}$$

Por hipótesis inductiva

$$\begin{aligned}
&= \int \mathcal{N}(x_{n+1}|\mu_{n+1}, \sigma_{n+1}^2) \mathcal{N}\left(x_1 + x_{n+1}|\mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2\right) dx_{n+1} \\
&= \int \mathcal{N}(x_{n+1}|\mu_{n+1}, \sigma_{n+1}^2) \mathcal{N}\left(x_{n+1}|-x_1 + \mu_t - \sum_{i>1}^n \mu_i, \sigma_t^2 + \sum_{i>1}^n \sigma_i^2\right) dx_{n+1} \\
&= \int \mathcal{N}(\mu_{n+1}|-x_1 + \mu_t - \sum_{i>1}^n \mu_i, \sigma_{n+1}^2 + \sigma_t^2 + \sum_{i>1}^n \sigma_i^2) \mathcal{N}(x_{n+1}|\mu_*, \sigma_*) dx_{n+1} \\
&= \mathcal{N}(\mu_{n+1}|-x_1 + \mu_t - \sum_{i>1}^n \mu_i, \sigma_{n+1}^2 + \sigma_t^2 + \sum_{i>1}^n \sigma_i^2) \underbrace{\int \mathcal{N}(x_{n+1}|\mu_*, \sigma_*) dx_{n+1}}_{=1} \tag{A.11} \\
&= \mathcal{N}(\mu_{n+1}|-x_1 + \mu_t - \sum_{i>1}^n \mu_i, \sigma_{n+1}^2 + \sigma_t^2 + \sum_{i>1}^n \sigma_i^2) \\
&= \mathcal{N}(x_1|\mu_t - \sum_{i>1}^n \mu_i - \mu_{n+1}, \sigma_{n+1}^2 + \sigma_t^2 + \sum_{i>1}^n \sigma_i^2) \\
&= \mathcal{N}(x_1|\mu_t - \sum_{i>1}^{n+1} \mu_i, \sigma_t^2 + \sum_{i>1}^{n+1} \sigma_i^2)
\end{aligned}$$

Luego, vale $P(n+1)$.

■ A.3 Multiplicación normales

■ A.3.1 Dos gaussianas

$$\mathcal{N}(s|\mu_1, \sigma_1^2) \mathcal{N}(s|\mu_2, \sigma_2^2) = \mathcal{N}\left(\frac{\sigma_1^2 \mu_2 + \sigma_2^2 \mu_1}{\sigma_1^2 + \sigma_2^2} \middle| \frac{1}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}\right) \tag{A.12}$$

A.3.2 Dos gaussianas de misma variable integradas

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathcal{N}(s|p_i, \beta^2) \mathcal{N}(s|\mu_i, \sigma_i^2) ds &= \int_{-\infty}^{\infty} \mathcal{N}(p|\mu_i, \beta^2 + \sigma_i^2) \mathcal{N}(s|\mu_*, \sigma_*^2) ds \\
&= \mathcal{N}(p|\mu_i, \beta^2 + \sigma_i^2) \underbrace{\int_{-\infty}^{\infty} \mathcal{N}(s|\mu_*, \sigma_*^2) ds}_{=1} \\
&= \mathcal{N}(p_i|\mu_i, \beta^2 + \sigma_i^2)
\end{aligned} \tag{A.13}$$

Dem:

$$\mathcal{N}(s|p_i, \beta^2) \mathcal{N}(s|\mu_i, \sigma_i^2) = \mathcal{N}(p|\mu_i, \beta^2 + \sigma_i^2) \mathcal{N}(s|\mu_*, \sigma_*^2) \tag{A.14}$$

$$\begin{aligned}
\mathcal{N}(s|p_i, \beta^2) \mathcal{N}(s|\mu_i, \sigma_i^2) &= \frac{1}{\sqrt{2\pi}\beta} e^{-\frac{1}{2}\left(\frac{(s-p_i)^2}{\beta^2}\right)} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{(s-\mu_i)^2}{\sigma_i^2}\right)} \\
&= \frac{1}{2\pi\beta\sigma_i} \underbrace{e^{-\frac{1}{2}\left(\frac{(s-p_i)^2}{\beta^2} + \frac{(s-\mu_i)^2}{\sigma_i^2}\right)}}_{e^{-\theta}}
\end{aligned} \tag{A.15}$$

$$\begin{aligned}
\theta &= \frac{(s-p)^2\sigma_i^2 + (s-\mu_i)^2\beta^2}{2\beta^2\sigma_i^2} = \frac{(\sigma_i^2 + \beta^2)s^2 - 2(p_i\sigma_i^2 + \mu_i\beta^2)s + (p_i^2\sigma_i^2 + \mu_i^2\beta^2)}{2\beta^2\sigma_i^2} \\
&= \frac{(s^2 - \frac{2(p_i\sigma_i^2 + \mu_i\beta^2)}{(\sigma_i^2 + \beta^2)}s + \frac{(p_i^2\sigma_i^2 + \mu_i^2\beta^2)}{(\sigma_i^2 + \beta^2)})}{\frac{2\beta^2\sigma_i^2}{(\sigma_i^2 + \beta^2)}}
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
\sigma_* &= \sqrt{\frac{\beta^2\sigma_i^2}{(\sigma_i^2 + \beta^2)}} \\
\mu_* &= \frac{(p\sigma_i^2 + \mu_i\beta^2)}{(\sigma_i^2 + \beta^2)}
\end{aligned} \tag{A.17}$$

$$\theta = \frac{s^2 - 2\mu_*s + \mu_*^2}{2\sigma_*^2} + \underbrace{\frac{\frac{(p_i^2\sigma_i^2 + \mu_i^2\beta^2)}{(\beta^2 + \sigma_i^2)} - \mu_*^2}{2\sigma_*}}_{\phi} \tag{A.18}$$

Nos queda entonces

$$\frac{1}{\beta\sigma_i} e^{-\frac{1}{2}\left(\frac{(s-\mu_*)^2}{2\sigma_*}\right)} e^{-\phi \sqrt{\frac{(\sigma_i^2 + \beta^2)}{(\sigma_i^2 + \beta^2)}}} = \frac{e^{-\phi}}{\sqrt{(\sigma_i^2 + \beta^2)}} \mathcal{N}(s|\mu_*, \sigma_*^2) \tag{A.19}$$

Trabajemos con ϕ

$$\begin{aligned}
\phi &= \frac{\frac{(p_i^2 \sigma_i^2 + \mu_i^2 \beta^2)}{(\beta^2 + \sigma_i^2)} - \mu_*^2}{2\sigma_*} \\
&= \frac{\frac{(p_i^2 \sigma_i^2 + \mu_i^2 \beta^2)}{(\beta^2 + \sigma_i^2)} - \frac{(p\sigma_i^2 + \mu_i \beta^2)^2}{(\sigma_i^2 + \beta^2)^2}}{2 \frac{\beta^2 \sigma_i^2}{(\sigma_i^2 + \beta^2)}} \\
&= \frac{(p_i^2 \sigma_i^2 + \mu_i^2 \beta^2)(\sigma_i^2 + \beta^2) - (p_i^2 \sigma_i^4 + 2p_i \mu_i \sigma_i^2 \beta^2 + \mu_i^2 \beta^4)}{2(\beta^2 + \sigma_i^2)} \frac{1}{2\beta^2 \sigma_i^2} \quad (\text{A.20}) \\
&= \frac{p_i^2 \sigma_i^4 + \mu_i^2 \beta^2 \sigma_i^2 + p_i^2 \sigma_i^2 \beta^2 + \mu_i^2 \beta^4 - p_i^2 \sigma_i^4 - 2p_i \mu_i \sigma_i^2 \beta^2 - \mu_i^2 \beta^4}{2(\beta^2 + \sigma_i^2)} \frac{1}{2\beta^2 \sigma_i^2} \\
&= \frac{+ \mu_i^2 \beta^2 \sigma_i^2 + p_i^2 \beta^2 \sigma_i^2 - 2p_i \mu_i \beta^2 \sigma_i^2}{2(\beta^2 + \sigma_i^2)} \frac{1}{2\beta^2 \sigma_i^2} \\
&= \frac{\mu_i^2 - 2p_i \mu_i + p_i^2}{2(\beta^2 + \sigma_i^2)} = \frac{(\mu_i - p_i)^2}{2(\beta^2 + \sigma_i^2)} = \phi
\end{aligned}$$

Si metemos ϕ

$$\begin{aligned}
\frac{1}{\sqrt{(\sigma_i^2 + \beta^2)}} e^{-\phi} \mathcal{N}(s|\mu_*, \sigma_*^2) &= \frac{e^{-\frac{(\mu_i - p_i)^2}{2(\beta^2 + \sigma_i^2)}}}{\sqrt{(\sigma_i^2 + \beta^2)}} \mathcal{N}(s|\mu_*, \sigma_*^2) \\
&= \underbrace{\mathcal{N}(\mu_i|p_i, \beta^2 + \sigma_i^2)}_{cte} \mathcal{N}(s|\mu_*, \sigma_*^2) \quad (\text{A.21}) \\
&= \mathcal{N}(p_i|\mu_i, \beta^2 + \sigma_i^2) \mathcal{N}(s|\mu_*, \sigma_*^2)
\end{aligned}$$

Apéndice B

Métodos

B.1 TrueSkill

Todo se basa en el pasaje de mensajes

$$m_{x \rightarrow f_x}(x) = \prod_{g \in n(x) \setminus \{f\}} m_{g \rightarrow x}(x) \quad (\text{B.1})$$

donde g son todos los factores que llegan al nodo x sin contar el factor al cual se le esta enviado.

$$m_{f_x \rightarrow x}(x) = \int \cdots \int \left(\prod_{k \in n(f) \setminus \{x\}} m_{k \rightarrow f}(x) \right) d\mathbf{x}_{\setminus x} \quad (\text{B.2})$$

donde k son todas las variables en contacto con el factor f , menos la variable x a la cual se esta enviando el mensaje.

B.1.1 Mensajes descendentes

$m_{f_{s_i} \rightarrow s_i}(s) :$

$$\begin{aligned} m_{f_{s_i} \rightarrow s_i}(s_i) &= \int \cdots \int f_{s_i}(\mathbf{x}) \prod_{k \in n(f_{s_i}) \setminus \{s_i\}} m_{k \rightarrow f_{s_i}}(k) d\mathbf{x}_{\setminus \{s_i\}} \\ &= \int \cdots \int f_{s_i}(s_i) d\mathbf{x}_{\setminus \{s_i\}} \\ &= \int \cdots \int \mathcal{N}(s_i | \mu_i, \sigma_i^2) d\mathbf{x}_{\setminus \{s_i\}} \\ &= \mathcal{N}(s_i | \mu_i, \sigma_i^2) \end{aligned} \quad (\text{B.3})$$

$m_{s \rightarrow f_p}(s) :$

$$m_{s_i \rightarrow f_{p_i}}(s_i) = \prod_{g \in n(s_i) \setminus \{f_{p_i}\}} m_{g \rightarrow s_i}(s_i) = m_{f_{s_i} \rightarrow s_i}(s_i) = \mathcal{N}(s_i | \mu_i, \sigma_i^2) \quad (\text{B.4})$$

$m_{f_p \rightarrow p}(p) :$

$$\begin{aligned}
m_{f_{p_i} \rightarrow p_i}(p_i) &= \int \cdots \int f_{p_i}(\mathbf{x}) \prod_{k \in n(f_{p_i}) \setminus \{p_i\}} m_{k \rightarrow f_{p_i}}(k) d\mathbf{x}_{\setminus \{p_i\}} \\
&= \int \cdots \int f_{p_i}(p_i) m_{s_i \rightarrow f_{p_i}}(s_i) d\mathbf{x}_{\setminus \{p_i\}} \\
&= \int \mathcal{N}(p_i | s_i, \beta_i^2) \mathcal{N}(s_i | \mu_i, \sigma_i^2) ds_i \\
&= \int \mathcal{N}(s_i | p_i, \beta_i^2) \mathcal{N}(s_i | \mu_i, \sigma_i^2) ds_i \\
&= \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2)
\end{aligned} \tag{B.5}$$

$m_{p \rightarrow f_t}(p) :$

$$\begin{aligned}
m_{p_i \rightarrow f_{t_e}}(p_i) &= \prod_{g \in n(p_i) \setminus \{f_{t_e}\}} m_{g \rightarrow p_i}(p_i) \\
&= m_{f_{p_i} \rightarrow p_i}(p_i) \\
&= \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2)
\end{aligned} \tag{B.6}$$

$m_{f_t \rightarrow t}(t) :$ t_e el sub indice describe a los equipos

$$\begin{aligned}
m_{f_{t_e} \rightarrow t_e}(t_e) &= \int \cdots \int f_{t_e}(\mathbf{x}) \prod_{k \in n(f_{t_e}) \setminus \{t_e\}} m_{k \rightarrow f_{t_e}}(k) d\mathbf{x}_{\setminus \{t_e\}} \\
&= \int \int f_{t_e}(t_e) m_{p_i \rightarrow f_{t_e}}(p_i) m_{p_j \rightarrow f_{t_e}}(p_j) dp_i dp_j \\
&= \int \int I(t_e = p_i + p_j) \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \mathcal{N}(p_j | \mu_j, \beta_j^2 + \sigma_j^2) dp_i dp_j \\
&= \int \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \mathcal{N}(t_e - p_i | \mu_j, \beta_j^2 + \sigma_j^2) dp_i \\
&= \int \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \mathcal{N}(-p_i | \mu_j - t_e, \beta_j^2 + \sigma_j^2) dp_i \\
&= \int \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \mathcal{N}(p_i | t_e - \mu_j, \beta_j^2 + \sigma_j^2) dp_i \\
&= \underbrace{\int \mathcal{N}(p_i | \mu_*, \sigma_*^2) dp_i}_{=1} \mathcal{N}(\mu_i | t_e - \mu_j, \beta_i^2 + \sigma_j^2 + \beta_j^2 + \sigma_i^2) \\
&= \mathcal{N}(-t_e | -\mu_i - \mu_j, \beta_i^2 + \beta_j^2 + \sigma_i^2 + \sigma_j^2) \\
&= \mathcal{N}(t_e | \mu_i + \mu_j, \beta_i^2 + \beta_j^2 + \sigma_i^2 + \sigma_j^2)
\end{aligned} \tag{B.7}$$

N Jugadores

$$m_{f_{t_e} \rightarrow t_e}(t_e) = \mathcal{N}(t_e | \underbrace{\sum_{i \in A_e} \mu_i}_{\mu_e}, \underbrace{\sum_{i \in A_e} (\beta_i^2 + \sigma_i^2)}_{\sigma_e^2}) = \mathcal{N}(t_e | \mu_e, \sigma_e^2) \tag{B.8}$$

$m_{t \rightarrow f_d}(t) :$

$$m_{t_e \rightarrow f_{d_k}}(d_k) = \prod_{g \in n(t_e) \setminus \{f_{d_k}\}} m_{g \rightarrow t_e}(t_e) = m_{f_{t_e} \rightarrow t_e}(t_e) = \mathcal{N}(t_e | \mu_e, \sigma_e^2) \tag{B.9}$$

$m_{f_d \rightarrow d}(d) :$

$$\begin{aligned}
m_{f_{d_1} \rightarrow d_1}(d_1) &= \int \cdots \int f_{d_1}(\mathbf{x}) \prod_{k \in n(f_{d_1}) \setminus \{d_1\}} m_{k \rightarrow f_{d_1}}(k) d\mathbf{x}_{\setminus \{d_1\}} \\
&= \int \int f_{d_1}(d_1) m_{t_a \rightarrow f_{d_1}}(t_a) m_{t_b \rightarrow f_{d_1}}(t_b) dt_a dt_b \\
&= \int \int I(d_1 = t_a - t_b) \mathcal{N}(t_a | \mu_a, \sigma_a^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_a dt_b \\
&= \int \mathcal{N}(d_1 + t_b | \mu_a, \sigma_a^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\
&= \int \mathcal{N}(t_b | \mu_a - d_1, \sigma_a^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\
&= \mathcal{N}(\mu_a - d_1 | \mu_b, \sigma_a^2 + \sigma_b^2) \\
&= \mathcal{N}(d_1 | \mu_a - \mu_b, \sigma_a^2 + \sigma_b^2)
\end{aligned} \tag{B.10}$$

Generalización

$$\begin{aligned}
m_{f_{d_1} \rightarrow d_1}(d_1) &= \mathcal{N}(d_1 | \underbrace{\sum_{i \in A_a} \mu_i - \sum_{i \in A_b} \mu_i}_{\delta}, \underbrace{\sum_{i \in A_a \cup A_b} (\beta_i^2 + \sigma_i^2)}_{\vartheta^2}) \\
&= \mathcal{N}(d_1 | \delta, \vartheta^2)
\end{aligned} \tag{B.11}$$

$m_{d \rightarrow f_r}(r) :$

$$m_{d \rightarrow f_r}(d) = \prod_{g \in n(t_e) \setminus \{f_{d_k}\}} m_{g \rightarrow t_e}(t_e) = m_{f_d \rightarrow d}(d) = \mathcal{N}(d_1 | \delta, \vartheta^2) \tag{B.12}$$

$m_{f_r \rightarrow r}(r) :$ (Caso ganador)

$$\begin{aligned}
m_{f_{r_1} \rightarrow r_1}(r_1) &= \int \cdots \int f_{r_1}(\mathbf{x}) \prod_{k \in n(f_{r_1}) \setminus \{r_1\}} m_{h \rightarrow f_{r_1}}(h) d\mathbf{x}_{\setminus \{r_1\}} \\
&= \int f_{r_1}(r_1) m_{d_1 \rightarrow f_{r_1}}(d_1) dd_1 \\
&= \int \mathbb{I}(d_1 > 0) \mathcal{N}(d_1 | \delta, \vartheta^2) dd_1 \\
&= \int_{-\infty}^0 \mathcal{N}(d_1 | -\delta, \vartheta^2) dd_1 \\
&= \frac{1}{\sqrt{2\pi}\vartheta} \int_{-\infty}^0 e^{-\frac{1}{2} \frac{(d_1 + \delta)^2}{\vartheta^2}} dd_1 \\
&= \frac{1}{\sqrt{2\pi}\vartheta} \int_{-\infty}^{\hat{x} = \frac{\delta}{\vartheta}} e^{-\frac{1}{2} x^2} dx \vartheta \\
&= \Phi\left(\frac{\delta}{\vartheta} | 0, 1\right)
\end{aligned} \tag{B.13}$$

B.1.2 Ascendentes

$m_{f_r \rightarrow d}(d) :$

$$m_{f_r \rightarrow d_1}(d_1) = I(d_1 > 0) \quad (\text{B.14})$$

$m_{d \rightarrow f_d}(d) :$

$$m_{d_1 \rightarrow f_{d_1}}(d_1) = \prod_{g \in n(d_1) \setminus \{f_{d_1}\}} m_{g \rightarrow d_1}(d_1) = m_{f_{r_1} \rightarrow d_1}(d_1) = I(d_1 > 0) \quad (\text{B.15})$$

$m_{f_{d_1} \rightarrow t_a}(t_a) :$ (caso ganador)

$$\begin{aligned} m_{f_{d_1} \rightarrow t_a}(t_a) &= \int \cdots \int f_{d_1}(\mathbf{x}) \prod_{k \in n(f_{d_1}) \setminus \{t_a\}} m_{k \rightarrow f_{d_1}}(k) d\mathbf{x}_{\setminus \{t_a\}} \\ &= \int \cdots \int f_{d_1}(d_1) m_{d_1 \rightarrow f_{d_1}}(d_1) m_{t_b \rightarrow f_{d_1}}(t_b) d\mathbf{x}_{\setminus \{t_a\}} \\ &= \int \cdots \int I(d_1 = t_a - t_b) I(d_1 > 0) \mathcal{N}(t_b | \mu_b, \sigma_b^2) d\mathbf{x}_{\setminus \{t_a\}} \\ &= \int \int I(d_1 = t_a - t_b) I(d_1 > 0) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dd_1 dt_b \\ &= \int I(t_a > t_b) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\ &= \Phi(t_a | \mu_b, \sigma_b^2) = \Phi(t_a | \sum_{i \in A_b} \mu_i, \sum_{i \in A_b} (\beta_i^2 + \sigma_i^2)) \end{aligned} \quad (\text{B.16})$$

$m_{f_{d_1} \rightarrow t_b}(t_b) :$ (Caso perdedor)

$$\begin{aligned} m_{f_{d_1} \rightarrow t_b}(t_b) &= \int \cdots \int f_{d_1}(\mathbf{x}) \prod_{k \in n(f_{d_1}) \setminus \{t_b\}} m_{k \rightarrow f_{d_1}}(k) d\mathbf{x}_{\setminus \{t_b\}} \\ &= \int \cdots \int f_{d_1}(d_1) m_{d_1 \rightarrow f_{d_1}}(d_1) m_{t_a \rightarrow f_{d_1}}(t_a) d\mathbf{x}_{\setminus \{t_b\}} \\ &= \int \cdots \int I(d_1 = t_a - t_b) I(d_1 > 0) \mathcal{N}(t_a | \mu_a, \sigma_a^2) d\mathbf{x}_{\setminus \{t_b\}} \\ &= \int \int I(d_1 = t_a - t_b) I(d_1 > 0) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dd_1 dt_a \\ &= \int I(t_a > t_b) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dt_a \\ &= \int_{t_b}^{\infty} \mathcal{N}(t_a | \mu_a, \sigma_a^2) dt_a \\ &= 1 - \Phi(t_b | \mu_a, \sigma_a^2) \\ &= 1 - \Phi(t_b | \sum_{i \in A_a} \mu_i, \sum_{i \in A_a} (\beta_i^2 + \sigma_i^2)) \end{aligned} \quad (\text{B.17})$$

$m_{t_a \rightarrow f_{t_a}}(t_a) :$ (Caso ganador)

$$\begin{aligned}
m_{t_a \rightarrow f_{t_a}}(t_a) &= \prod_{g \in n(t_a) \setminus \{f_{t_a}\}} m_{g \rightarrow t_a}(t_a) \\
&= m_{f_{d_1} \rightarrow t_a}(t_a) = \Phi(t_a | \mu_b, \sigma_b^2)
\end{aligned} \tag{B.18}$$

$m_{t_b \rightarrow f_{t_b}}(t_b) :$ (Caso perdedor)

$$\begin{aligned}
m_{t_b \rightarrow f_{t_b}}(t_b) &= \prod_{g \in n(t_b) \setminus \{f_{t_b}\}} m_{g \rightarrow t_b}(t_b) \\
&= m_{f_{d_1} \rightarrow t_b}(t_b) = 1 - \Phi(t_b | \mu_a, \sigma_a^2)
\end{aligned} \tag{B.19}$$

$m_{f_{t_a} \rightarrow p_1}(p_1) :$ (Caso ganador)

$$\begin{aligned}
m_{f_{t_a} \rightarrow p_1}(p_1) &= \int \cdots \int f_{t_a}(\mathbf{x}) \prod_{k \in n(f_{t_a}) \setminus \{p_1\}} m_{k \rightarrow f_{t_a}}(k) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \int \cdots \int f_{t_a}(t_a) m_{p_2 \rightarrow f_{t_a}}(p_2) m_{t_a \rightarrow f_{t_a}}(t_a) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \int \cdots \int \mathbb{I}(t_a = p_1 + p_2) \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \Phi(t_a | \mu_b, \sigma_b^2) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \iint \mathbb{I}(t_a = p_1 + p_2) \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \Phi(t_a | \mu_b, \sigma_b^2) dt_a dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \Phi(p_1 + p_2 | \mu_b, \sigma_b^2) dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \Phi(p_1 | \mu_b - p_2, \sigma_b^2) dp_2 \\
&= \kappa(p_1)
\end{aligned} \tag{B.20}$$

La derivada de la función de distribución acumulada $\Phi(\cdot)$ es el valor de la densidad de la función de probabilidad $\mathcal{N}(\cdot)$. Con esta idea en mente, tomamos la derivada de ambos lados de la igualdad:

$$\begin{aligned}
\frac{\partial \kappa(p_1)}{\partial p_1} &= \frac{\partial}{\partial p_1} \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \Phi(p_1 | \mu_b - p_2, \sigma_b^2) dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \frac{\partial}{\partial p_1} \Phi(p_1 | \mu_b - p_2, \sigma_b^2) dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \mathcal{N}(p_1 | \mu_{p_1} - p_2, \sigma_b^2) dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_2, \beta_2^2 + \sigma_2^2) \mathcal{N}(p_2 | \mu_b - p_1, \sigma_b^2) dp_2 \\
&= \mathcal{N}(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2)
\end{aligned} \tag{B.21}$$

Como $m_{f_{t_a} \rightarrow p_1}(p_1) = \kappa(p_1)$ y $\frac{\partial \kappa(p_1)}{\partial p_1} = \mathcal{N}(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2)$, usando la propiedad $\frac{\partial \Phi(x | \mu, \sigma^2)}{\partial x} = \mathcal{N}(x | \mu, \sigma^2)$ Entonces, $m_{f_{t_a} \rightarrow p_1}(p_1) = \Phi(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2)$

$$m_{f_{t_a} \rightarrow p_1}(p_1) = \Phi(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2) = \Phi\left(p_1 \mid \sum_{i \in A_b} \mu_i - \mu_2, \beta_2^2 + \sigma_2^2 + \sum_{i \in A_b} \beta_i^2 + \sigma_i^2\right) \tag{B.22}$$

$m_{f_{t_b} \rightarrow p_3}(p_3)$: (Caso perdedor)

$$\begin{aligned}
m_{f_{t_b} \rightarrow p_3}(p_3) &= \int \cdots \int f_{t_a}(\mathbf{x}) \prod_{k \in n(f_{t_a}) \setminus \{p_1\}} m_{k \rightarrow f_{t_a}}(k) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \int \cdots \int f_{t_a}(t_a) m_{t_b \rightarrow f_{t_b}}(t_b) m_{p_4 \rightarrow f_{t_b}}(p_4) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \int \cdots \int \mathbb{I}(t_b = p_3 + p_4) (1 - \Phi(t_b | \mu_a, \sigma_a^2)) \mathcal{N}(p_4 | \mu_4, \beta_4^2 + \sigma_4^2) d\mathbf{x}_{\setminus \{p_3\}} \\
&= \iint \mathbb{I}(t_b = p_3 + p_4) \mathcal{N}(p_4 | \mu_4, \beta_4^2 + \sigma_4^2) (1 - \Phi(t_b | \mu_a, \sigma_a^2)) dt_b dp_4 \\
&= \int \mathcal{N}(p_4 | \mu_4, \beta_4^2 + \sigma_4^2) (1 - \Phi(p_3 | \mu_a - p_4, \sigma_a^2)) dp_4 \\
&= \underbrace{\int \mathcal{N}(p_4 | \mu_4, \beta_4^2 + \sigma_4^2) dp_4}_1 - \underbrace{\int \mathcal{N}(p_4 | \mu_4, \beta_4^2 + \sigma_4^2) \Phi(p_3 | \mu_a - p_4, \sigma_a^2) dp_4}_{\kappa(p_3)} \\
&= 1 - \Phi(p_3, \mu_a - \mu_4, \beta_4^2 + \sigma_4^2 + \sigma_a^2) \\
&= 1 - \Phi\left(p_3, \sum_{i \in A_a} \mu_i - \mu_4, \beta_4^2 + \sigma_4^2 + \sum_{i \in A_a} \beta_i^2 + \sigma_i^2\right)
\end{aligned} \tag{B.23}$$

$m_{p_1 \rightarrow f_{p_1}}(s_1)$: (Caso ganador)

$$\begin{aligned}
m_{p_1 \rightarrow f_{p_1}}(p_1) &= \prod_{g \in n(p_1) \setminus \{f_{p_1}\}} m_{g \rightarrow p_1}(p_1) \\
&= m_{f_{t_a} \rightarrow p_1}(p_1) = \Phi(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2)
\end{aligned} \tag{B.24}$$

$m_{p_3 \rightarrow f_{p_3}}(s_1)$: (Caso perdedor)

$$\begin{aligned}
m_{p_3 \rightarrow f_{p_3}}(p_3) &= \prod_{g \in n(p_3) \setminus \{f_{p_3}\}} m_{g \rightarrow p_3}(p_3) \\
&= m_{f_{t_b} \rightarrow p_3}(p_3) = 1 - \Phi(p_3, \mu_a - \mu_4, \beta_4^2 + \sigma_4^2 + \sigma_a^2)
\end{aligned} \tag{B.25}$$

$m_{f_{p_1} \rightarrow s_1}(s_1)$: (Caso ganador)

$$\begin{aligned}
m_{f_{p_1} \rightarrow s_1}(s_1) &= \int \cdots \int f_{p_1}(\mathbf{x}) \prod_{h \in n(f_{p_1}) \setminus \{s_1\}} m_{h \rightarrow f_{p_1}}(h) d\mathbf{x}_{\setminus \{s_1\}} \\
&= \int \cdots \int f_{p_1}(p_1) m_{p_1 \rightarrow f_{p_1}}(p_1) d\mathbf{x}_{\setminus \{s_1\}} \\
&= \int \cdots \int \mathcal{N}(p_1 | s_1, \beta_1^2) \Phi(p_1 | \mu_b - \mu_2, \beta_2^2 + \sigma_2^2 + \sigma_b^2) d\mathbf{x}_{\setminus \{s_1\}} \\
&= \int \mathcal{N}(p_1 | s_1, \beta_1^2) \Phi(\mu_2 | \mu_b - p_1, \beta_2^2 + \sigma_2^2 + \sigma_b^2) dp_1 \\
&= \Phi(s_1 | \mu_b - \mu_2, \beta_1^2 + \beta_2^2 + \sigma_2^2 + \sigma_b^2) = \Phi(s_1 | \mu_b - \mu_a + \mu_1, \sigma_b^2 + \sigma_a^2 - \sigma_1^2)
\end{aligned} \tag{B.26}$$

General (N vs N)

$$\begin{aligned}
m_{f_{p_1} \rightarrow s_1}(s_1) &= \Phi(s_1 | \mu_b - \mu_a + \mu_1, \sigma_b^2 + \sigma_a^2 - \sigma_1^2) \\
&= \Phi\left(s_1 \mid \underbrace{\sum_{i \in A_b} \mu_i - \sum_{i \in A_a} \mu_i}_{-\delta} + \mu_1, \underbrace{\sum_{i \in A_b \cup A_a} (\beta_i^2 + \sigma_i^2)}_{\vartheta^2} - \sigma_1^2\right) \\
&= \Phi(s_1 | -\delta + \mu_1, \vartheta^2 - \sigma_1^2) \\
&= 1 - \Phi(-s_1 | \delta - \mu_1, \vartheta^2 - \sigma_1^2) \\
&= 1 - \Phi(0 | \delta - \mu_1 + s_1, \vartheta^2 - \sigma_1^2) \\
&= 1 - \Phi(0 | \delta_1(s_1), \vartheta_1^2) \\
&= 1 - \Phi\left(\frac{0 - \delta_1(s_1)}{\vartheta_1} \mid 0, 1\right) \\
&= \Phi\left(\frac{\delta_1(s_1)}{\vartheta_1} \mid 0, 1\right)
\end{aligned} \tag{B.27}$$

Nota: el mensaje $m_{f_{p_1} \rightarrow s_1}(s_1)$ computa la “probabilidad de ganar parametrizada”, esto es la probabilidad de ganar si conociéramos la habilidad del jugador. Si conocemos la habilidad del jugador entonces hay que eliminar la varianza de la distribución de creencias de la varianza total (lo que hacemos en ϑ_1) y hay que remplazar la media de la distribución de creencias por la verdadera habilidad en la diferencia esperada (lo que hacemos en $\delta_1(s_1)$).

$m_{f_{p_3} \rightarrow s_3}(s_3)$: (Caso perdedor)

$$\begin{aligned}
m_{f_{p_3} \rightarrow s_3}(s_3) &= \int \cdots \int f_{p_3}(\mathbf{x}) \prod_{h \in n(f_{p_3}) \setminus \{s_3\}} m_{h \rightarrow f_{p_3}}(h) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \cdots \int f_{p_3}(p_3) m_{p_3 \rightarrow f_{p_3}}(p_3) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \cdots \int \mathcal{N}(p_3 | s_3, \beta_3^2) (1 - \Phi(p_3, \mu_a - \mu_4, \beta_4^2 + \sigma_4^2 + \sigma_a^2)) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \mathcal{N}(p_3 | s_3, \beta_3^2) (1 - \Phi(\mu_4, \mu_a - p_3, \beta_4^2 + \sigma_4^2 + \sigma_a^2)) dp_3 \\
&= \int \mathcal{N}(p_3 | s_3, \beta_3^2) dp_3 - \int \mathcal{N}(p_3 | s_3, \beta_3^2) \Phi(\mu_4, \mu_a - p_3, \beta_4^2 + \sigma_4^2 + \sigma_a^2) dp_3 \\
&= 1 - \Phi\left(s_3 | \mu_a - \mu_4, \beta_3^2 + \beta_4^2 + \sigma_4^2 + \sigma_a^2\right) \\
&= 1 - \Phi\left(s_3 | \mu_a - \mu_4, \sigma_b^2 - \sigma_3^2 + \sigma_a^2\right)
\end{aligned} \tag{B.28}$$

General (N vs N)

$$\begin{aligned}
m_{f_{p_3} \rightarrow s_3}(s_3) &= 1 - \Phi(s_3 | \underbrace{\mu_a - \mu_b}_{\delta} + \mu_3, \underbrace{\sigma_a^2 + \sigma_b^2}_{\vartheta^2} - \sigma_3^2) \\
&= 1 - \Phi(s_3 | \delta + \mu_3, \vartheta^2 - \sigma_3^2) = \Phi(0 | \underbrace{-\delta - \mu_3 + s_3}_{\delta_3(s_3)}, \underbrace{\vartheta^2 - \sigma_3^2}_{\vartheta_3^2}) \\
&= \Phi(0 | \delta_3(s_3), \vartheta_3^2) = \Phi\left(\frac{0 - \delta_3(s_3)}{\vartheta_3} | 0, 1\right) \\
&= \Phi\left(\frac{-\delta_3(s_3)}{\vartheta_3} | 0, 1\right)
\end{aligned} \tag{B.29}$$

Siendo $s_i \in t_i$. Si $t_i = a$ entonces $t_j = b$ y viceversa.

$$\delta = \mu_{win} - \mu_{lose} \tag{B.30}$$

Equipo t_i gana.

$$\delta_i(s_i) = \delta - \mu_i + s_i = \mu_{t_i} - \mu_{t_j} - \mu_i + s_i \tag{B.31}$$

Equipo t_j pierde.

$$\delta_j(s_j) = -\delta - \mu_j + s_j = \mu_{t_j} - \mu_{t_i} - \mu_j + s_j \tag{B.32}$$

$$\vartheta^2 = \sigma_{t_i}^2 + \sigma_{t_j}^2 \tag{B.33}$$

$$\vartheta_i^2 = \vartheta^2 - \sigma_i^2 = \sigma_{t_i}^2 + \sigma_{t_j}^2 - \sigma_i^2 \tag{B.34}$$

Nota: el mensaje $m_{f_{p_3} \rightarrow s_3}(s_3)$ computa la “probabilidad de perder parametrizada” (que es la misma que la probabilidad de ganar de su contrincante). Si conociéramos la habilidad del jugador entonces hay que eliminar la varianza de la distribución de creencias de la varianza total (lo que hacemos en ϑ_3) y hay que remplazar la media de la distribución de creencias por la verdadera habilidad en la diferencia esperada (lo que hacemos en $\delta_3(s_3)$).

■ B.1.3 Posterior ganador y perdedor

Si se realiza, estos mensajes para calcular la posterior, este deja de ser una gaussiana si no lo que se conoce como gaussiana truncada.

Ganador

$$p(s_1 | o, A) = \prod_{h \in n(x_i)} m_{h \rightarrow x_i} = \mathcal{N}(s_1 | \mu_1, \sigma_1^2) \Phi\left(\frac{\delta_1(s_1)}{\vartheta_1}\right) \tag{B.35}$$

Normal truncada Veamos el mensaje que genera que el resto de los mensajes dejen de ser gaussianos. Para ello veamos la marginal del nodo d_1 .

$$\text{Rec: } \delta = \sum_{i \in A_{win}} \mu_i - \sum_{i \in A_{lose}} \mu_i = \mu_{win} - \mu_{lose}$$

$$\vartheta^2 = \sum_{i \in A_{win} \cup A_{lose}} (\beta_i^2 + \sigma_i^2)$$

$$\begin{aligned}
p(d_1) &= \prod_{h \in n(d_1)} m_{h \rightarrow d_1} = m_{f_{d_1} \rightarrow d_1}(d_1) m_{f_r \rightarrow d_1}(d_1) = m_{f_{d_1} \rightarrow d_1}(d_1) m_{d_1 \rightarrow f_{d_1}}(d_1) \\
&= \mathcal{N}(d_1 | \delta, \vartheta) \mathbb{I}(d_1 > 0)
\end{aligned} \tag{B.36}$$

Para tener una *posterior* normal lo que se hace es un buscar la normal que más se aproxima a esta normal truncada. La función de densidad de probabilidad, para una gaussiana truncada entre x_1 y x_2 , vale cero afuera del intervalo y dentro vale:

$$f(x_1 < x < x_2 | \delta, \vartheta, x_1, x_2) = \frac{1}{\vartheta} \frac{\mathcal{N}\left(\frac{x-\delta}{\vartheta} | 0, 1\right)}{\Phi\left(\frac{x_2-\delta}{\vartheta}\right) - \Phi\left(\frac{x_1-\delta}{\vartheta}\right)} \tag{B.37}$$

Con

$$\mathcal{N}(y | 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \tag{B.38}$$

$$\Phi(y | 0, 1) = \frac{1}{2}(1 + \operatorname{erf}(x/\sqrt{2})) \tag{B.39}$$

Por definición, si $x_2 = \infty$ entonces $\Phi\left(\frac{x_2-\delta}{\vartheta} | 0, 1\right) = 1$

Para ello, veamos la media o esperanza de dicha función. Calculando el primer momento de la distribución, obtenemos la media como,

$$\begin{aligned}
E(X | x_1 < X < x_2) &= \int_{x_1}^{x_2} p(d_1) s_1 ds_1 = \int_{x_1}^{x_2} \mathcal{N}(d_1 | \delta, \vartheta) \mathbb{I}(d_1 > 0) s_1 ds_1 \\
&= \delta + \vartheta \frac{\mathcal{N}\left(\frac{x_1-\delta}{\vartheta} | 0, 1\right) - \mathcal{N}\left(\frac{x_2-\delta}{\vartheta} | 0, 1\right)}{\Phi\left(\frac{x_2-\delta}{\vartheta}\right) - \Phi\left(\frac{x_1-\delta}{\vartheta}\right)} \\
&= \delta + \vartheta \frac{\mathcal{N}(\alpha | 0, 1) - \mathcal{N}(\beta | 0, 1)}{\Phi(\beta) - \Phi(\alpha)} \\
&= \delta + \vartheta \frac{\mathcal{N}(\alpha | 0, 1) - 0}{1 - \Phi(\alpha)} = \delta + \vartheta \frac{\mathcal{N}(\alpha | 0, 1)}{1 - \Phi(\alpha)}
\end{aligned} \tag{B.40}$$

Con $\alpha = \frac{x_1-\delta}{\vartheta} = -\frac{\delta}{\vartheta}$ y $\beta = \frac{x_2-\delta}{\vartheta} = \infty$.

La varianza se define mediante el segundo momento de la distribución como:

$$\begin{aligned}
V(X | x_1 < X < x_2) &= \vartheta^2 \left(1 + \left(\frac{\alpha \mathcal{N}(\alpha | 0, 1) - \beta \mathcal{N}(\beta | 0, 1)}{\Phi(\beta | 0, 1) - \Phi(\alpha | 0, 1)} \right) - \left(\frac{\mathcal{N}(\alpha | 0, 1) - \mathcal{N}(\beta | 0, 1)}{\Phi(\beta | 0, 1) - \Phi(\alpha | 0, 1)} \right)^2 \right) \\
&= \vartheta^2 \left(1 + \left(\frac{\alpha \mathcal{N}(\alpha | 0, 1) - \beta 0}{1 - \Phi(\alpha | 0, 1)} \right) - \left(\frac{\mathcal{N}(\alpha | 0, 1) - 0}{1 - \Phi(\alpha | 0, 1)} \right)^2 \right) \\
&= \vartheta^2 \left(1 + \left(\frac{\alpha \mathcal{N}(\alpha | 0, 1)}{1 - \Phi(\alpha | 0, 1)} \right) - \left(\frac{\mathcal{N}(\alpha | 0, 1)}{1 - \Phi(\alpha | 0, 1)} \right)^2 \right)
\end{aligned} \tag{B.41}$$

$\mathcal{N}(\beta | 0, 1)$ decae más rapido que lo que β crece ya que esta primera es exponencial negativa con mismo argumento que el otro que es lineal.

$$\begin{aligned}
E(X|X > 0) &= \delta + \vartheta \frac{\mathcal{N}(\alpha|0, 1)}{1 - \Phi(\alpha|0, 1)} = \delta + \vartheta \frac{\mathcal{N}(\frac{x_1 - \delta}{\vartheta}|0, 1)}{1 - \Phi(\frac{0 - \delta}{\vartheta}|0, 1)} \\
&= \delta + \vartheta \underbrace{\frac{\mathcal{N}(\frac{\delta - 0}{\vartheta}|0, 1)}{\Phi(\frac{\delta - 0}{\vartheta}|0, 1)}}_{V_1(t)} = \delta + \vartheta V_1(t)
\end{aligned} \tag{B.42}$$

$$V_1(t) = \frac{\mathcal{N}(\frac{\delta}{\vartheta}|0, 1)}{\Phi(\frac{\delta}{\vartheta}|0, 1)} \tag{B.43}$$

con $t = \frac{\delta - x_1}{\vartheta} = -\alpha$

Nos interesa el caso de $x_1 = 0$. Notacion $\mathcal{N}(\alpha) = \mathcal{N}(\alpha|0, 1)$ y $\Phi(\alpha) = \Phi(\alpha|0, 1)$

$$\begin{aligned}
V(X|X > a) &= \vartheta^2 \left(1 + \left(\frac{\alpha \mathcal{N}(\alpha)}{1 - \Phi(\alpha)} \right) - \left(\frac{\mathcal{N}(\alpha)}{1 - \Phi(\alpha)} \right)^2 \right) \\
&= \vartheta^2 \left(1 + \left(\frac{\alpha \mathcal{N}(-\alpha)}{\Phi(-\alpha)} \right) - \left(\frac{\mathcal{N}(-\alpha)}{\Phi(-\alpha)} \right)^2 \right) \\
&= \vartheta^2 \left(1 + \left(\frac{-t \mathcal{N}(t)}{\Phi(t)} \right) - \left(\frac{\mathcal{N}(t)}{\Phi(t)} \right)^2 \right) \\
&= \vartheta^2 (1 + -t V_1(t) - V_1(t)^2) \\
&= \vartheta^2 (1 + V_1(t)(-t - V_1(t))) \\
&= \vartheta^2 (1 - \underbrace{V_1(t)(V_1(t) + t)}_{W_1(t)}) = \vartheta^2 (1 - W_1(t))
\end{aligned} \tag{B.44}$$

$$\begin{aligned}
W_1(t) &= V_1(t)(V_1(t) + t) = \frac{\mathcal{N}(\frac{\delta - x_1}{\vartheta})}{\Phi(\frac{\delta - x_1}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta - x_1}{\vartheta})}{\Phi(\frac{\delta - x_1}{\vartheta})} + \frac{\delta - x_1}{\vartheta} \right) \\
&= \frac{\mathcal{N}(\frac{\delta - 0}{\vartheta})}{\Phi(\frac{\delta - 0}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta - 0}{\vartheta})}{\Phi(\frac{\delta - 0}{\vartheta})} + \frac{\delta - 0}{\vartheta} \right) \\
&= \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right)
\end{aligned} \tag{B.45}$$

Luego, la normal aproximada es

$$\hat{p}(d_1) = \mathcal{N}\left(d_1 \mid \delta + \vartheta V_1(t), \vartheta^2(1 - W_1(t))\right) \tag{B.46}$$

donde, en caso de que no se contemple el empate, $\alpha = \frac{-\delta}{\vartheta}$.

Teniendo la normal aproximada $\hat{p}(d_1)$ podemos calcular el mensaje ascendentes aproximado. Para ello se usa el "truco" de calcular la marginal de ese nodo, y sacarle el mensaje correspondiente.

$$\begin{aligned}
\hat{\delta} &= \delta + \vartheta V_1(t) \\
&= \mu_a - \mu_b + \sqrt{\sum_{i \in A_a \cup A_b} (\beta_i^2 + \sigma_i^2)} \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})}
\end{aligned} \tag{B.47}$$

$$\begin{aligned}
\hat{\vartheta}^2 &= \vartheta^2(1 - W_1(t)) = \\
&= \left[\sum_{i \in A_a \cup A_b} (\beta_i^2 + \sigma_i^2) \right] \left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right]
\end{aligned} \tag{B.48}$$

$$\hat{m}_{d \rightarrow f_d(d)}$$

$$\begin{aligned}
m_{d_1 \rightarrow f_{d_1}}(d_1) &= \frac{p(d_1)}{m_{f_{d_1} \rightarrow d_1}(d_1)} \\
&\approx \frac{\hat{p}(d_1)}{m_{f_{d_1} \rightarrow d_1}(d_1)} \\
&= \frac{\mathcal{N}(d_1 | \hat{\delta}, \hat{\vartheta}^2)}{\mathcal{N}(d_1 | \delta, \vartheta^2)} \\
&\propto \mathcal{N}(d_1, \delta_{\div}, \vartheta_{\div}^2)
\end{aligned} \tag{B.49}$$

$$\begin{aligned}
\delta_{\div} &= \frac{(\vartheta^2 \hat{\delta} - \hat{\vartheta}^2 \delta)}{(\vartheta^2 - \hat{\vartheta}^2)} = \frac{\left(\cancel{\vartheta^2} \left[\delta + \vartheta \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \right] \right)}{\left(\cancel{\vartheta^2} - \cancel{\vartheta^2} \left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right] \right)} \\
&\quad - \frac{\left(\cancel{\vartheta^2} \left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right] \delta \right)}{\left(\cancel{\vartheta^2} - \cancel{\vartheta^2} \left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right] \right)} \\
&= \frac{\left(\left[\delta + \vartheta \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \right] \right) - \left(\left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right] \delta \right)}{\left(1 - \left[1 - \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right] \right)} \\
&= \frac{\cancel{\delta} + \vartheta \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} - \cancel{\delta} + \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \delta}{\left(\cancel{\mathcal{I}} - \cancel{\mathcal{I}} + \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right)} \\
&= \frac{\cancel{\vartheta \frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})}} + \cancel{\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})}} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \delta}{\left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \right)} = \frac{\vartheta + \left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right) \delta}{\left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right)} \\
&= \frac{\vartheta}{\left(\frac{\mathcal{N}(\frac{\delta}{\vartheta})}{\Phi(\frac{\delta}{\vartheta})} + \frac{\delta}{\vartheta} \right)} + \delta = \frac{\vartheta}{(V_1(t) + \frac{\delta}{\vartheta})} + \delta
\end{aligned} \tag{B.50}$$

$$\begin{aligned}
\vartheta_{\div} &= \sqrt{\frac{\hat{\vartheta}^2 \vartheta^2}{(\vartheta^2 - \hat{\vartheta}^2)}} = \sqrt{\frac{\vartheta^2(1-W)\vartheta^2}{\vartheta^2 - \vartheta^2(1-W)}} \\
&= \sqrt{\frac{\vartheta^4(1-W)}{\vartheta^2(1 - (1-W))}} \\
&= \sqrt{\frac{\vartheta^4(1-W)}{\vartheta^2(W)}} \\
&= \sqrt{\frac{\vartheta^2(1-W)}{W}} \\
&= \vartheta \sqrt{\frac{(1-W)}{W}}
\end{aligned} \tag{B.51}$$

$\hat{m}_{f_d \rightarrow t_a}(t_a)$ (Caso ganador)

$$\begin{aligned}
m_{f_{d_1} \rightarrow t_a}(t_a) &= \int \cdots \int f_{d_1}(\mathbf{x}) \prod_{h \in n(f_{d_1}) \setminus \{t_a\}} m_{h \rightarrow f_{d_1}}(h) d\mathbf{x}_{\setminus \{t_a\}} \\
&= \int \cdots \int f_{d_1}(d_1) m_{d_1 \rightarrow f_{d_1}}(d_1) m_{t_b \rightarrow f_{d_1}}(t_b) d\mathbf{x}_{\setminus \{t_a\}} \\
&\approx \int \cdots \int \mathbb{I}(d_1 = t_a - t_b) \mathcal{N}(d_1 | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) d\mathbf{x}_{\setminus \{t_a\}} \\
\hat{m}_{f_{d_1} \rightarrow t_a}(t_a) &= \int \int \mathbb{I}(d_1 = t_a - t_b) \mathcal{N}(d_1 | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dd_1 dt_b \\
&= \int \mathcal{N}(t_a - t_b | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\
&= \int \mathcal{N}(-t_b | \delta_{\div} - t_a, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\
&= \int \mathcal{N}(t_b | t_a - \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_b | \mu_b, \sigma_b^2) dt_b \\
&= \mathcal{N}(t_a - \delta_{\div} | \mu_b, \vartheta_{\div}^2 + \sigma_b^2) \\
&= \mathcal{N}(t_a | \mu_b + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2)
\end{aligned} \tag{B.52}$$

$\hat{m}_{f_d \rightarrow f_{t_b}}(t_b)$ (Caso perdedor)

$$\begin{aligned}
m_{f_{d_1} \rightarrow t_b}(t_b) &= \int \cdots \int f_{d_1}(\mathbf{x}) \prod_{h \in n(f_{d_1}) \setminus \{t_b\}} m_{h \rightarrow f_{d_1}}(h) d\mathbf{x}_{\setminus \{t_b\}} \\
&= \int \cdots \int f_{d_1}(d_1) m_{d_1 \rightarrow f_{d_1}}(d_1) m_{t_a \rightarrow f_{d_1}}(t_a) d\mathbf{x}_{\setminus \{t_b\}} \\
&\approx \int \cdots \int \mathbb{I}(d_1 = t_a - t_b) \mathcal{N}(d_1 | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_a | \mu_a, \sigma_a^2) d\mathbf{x}_{\setminus \{t_a\}} \\
\hat{m}_{f_{d_1} \rightarrow t_b}(t_b) &= \int \int \mathbb{I}(d_1 = t_a - t_b) \mathcal{N}(d_1 | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dd_1 dt_a \quad (\text{B.53}) \\
&= \int \mathcal{N}(t_a - t_b | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dt_a \\
&= \int \mathcal{N}(t_a | \delta_{\div} + t_b, \vartheta_{\div}^2) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dt_a \\
&= \mathcal{N}(t_b + \delta_{\div} | \mu_a, \vartheta_{\div}^2 + \sigma_a^2) \\
&= \mathcal{N}(t_b | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2)
\end{aligned}$$

$\hat{m}_{f_{t_a} \rightarrow p_1}(p_1)$ (Caso ganador)

$$\begin{aligned}
m_{f_{t_a} \rightarrow p_1}(p_1) &= \int \cdots \int f_{t_a}(\mathbf{x}) \prod_{h \in n(f_{t_a}) \setminus \{p_1\}} m_{h \rightarrow f_{t_a}}(h) d\mathbf{x}_{\setminus \{p_1\}} \\
&= \int \cdots \int f_{t_a}(t_a) m_{f_{d_1} \rightarrow f_{t_a}}(t_a) m_{p_2 \rightarrow f_{t_a}}(p_2) d\mathbf{x}_{\setminus \{p_1\}} \\
&\approx \int \cdots \int \mathbb{I}(t_a = p_1 + p_2) \mathcal{N}(t_a | \mu_b + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta_2^2) d\mathbf{x}_{\setminus \{p_1\}} \\
\hat{m}_{f_{t_a} \rightarrow p_1}(p_1) &= \int \int \mathbb{I}(t_a = p_1 + p_2) \mathcal{N}(t_a | \mu_b + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta_2^2) dt_a dp_2 \\
&= \int \mathcal{N}(p_1 + p_2 | \mu_b + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta_2^2) dp_2 \\
&= \int \mathcal{N}(p_2 | \mu_b - p_1 + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta_2^2) dp_2 \\
&= \mathcal{N}(\mu_b - p_1 + \delta_{\div} | \mu_2, \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2 + \beta_2^2) \\
&= \mathcal{N}(p_1 | (\mu_b - \mu_2) + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2 + \beta_2^2) \quad (\text{B.54})
\end{aligned}$$

La generalización, se consigue utilizando la propiedad de suma de n normales, que se muestra en el anexo.

$$\begin{aligned}
\hat{m}_{f_{t_a} \rightarrow p_1}(p_1) &= \int \int \mathbb{I}(t_a = p_1 + p_2 + \dots + p_N) \mathcal{N}(t_a | \mu_b + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2) \mathcal{N}(p_2 | \mu_2, \sigma_2^2 + \beta_2^2) \\
&\quad \mathcal{N}(p_3 | \mu_2, \sigma_3^2 + \beta_3^2) \dots \mathcal{N}(p_N | \mu_N, \sigma_N^2 + \beta_N^2) dt_a dp_2 dp_3 \dots dp_N \\
&= \mathcal{N}(p_1 | (\mu_b - \sum_{i \in A_a, i \neq 1} \mu_i) + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2 + \sum_{i \in A_a, i \neq 1} \sigma_i^2 + \beta_i^2) \\
&= \mathcal{N}(p_1 | (\mu_b - (\mu_a - \mu_1)) + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2 + \underbrace{\sum_{i \in A_a, i \neq 1} \sigma_i^2 + \beta_i^2}_{\sigma_a^2 - \sigma_1^2 - \beta_1^2}) \\
&= \mathcal{N}(p_1 | (-\delta + \mu_1) + \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2 - \beta_1^2)) \quad (\text{B.55})
\end{aligned}$$

Donde en este caso, todos los jugadores son del mismo equipo t_a

$\hat{m}_{f_{t_b} \rightarrow p_3}(p_3)$ (Caso perdedor)

$$\begin{aligned}
m_{f_{t_b} \rightarrow p_3}(p_3) &= \int \cdots \int f_{t_b}(\mathbf{x}) \prod_{h \in n(f_{t_b}) \setminus \{p_3\}} m_{h \rightarrow f_{t_b}}(h) d\mathbf{x}_{\setminus \{p_3\}} \\
&= \int \cdots \int f_{t_b}(t_b) m_{f_{d_1} \rightarrow f_{t_b}}(t_b) m_{p_4 \rightarrow f_{t_b}}(p_4) d\mathbf{x}_{\setminus \{p_1\}} \\
&\approx \int \cdots \int \mathbb{I}(t_b = p_3 + p_4) \mathcal{N}(t_b | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_4 | \mu_4, \sigma_4^2 + \beta_4^2) d\mathbf{x}_{\setminus \{p_3\}} \\
\hat{m}_{f_{t_b} \rightarrow p_3}(p_3) &= \int \int \mathbb{I}(t_b = p_3 + p_4) \mathcal{N}(t_b | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_4 | \mu_4, \sigma_4^2 + \beta_4^2) dt_b dp_4 \\
&= \int \mathcal{N}(p_3 + p_4 | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_4 | \mu_4, \sigma_4^2 + \beta_4^2) dp_4 \\
&= \int \mathcal{N}(p_4 | \mu_a - p_3 - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_4 | \mu_4, \sigma_4^2 + \beta_4^2) dp_4 \\
&= \mathcal{N}(\mu_a - p_3 - \delta_{\div} | \mu_4, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2 + \beta_4^2) \\
&= \mathcal{N}(p_3 | (\mu_a - \mu_4) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2 + \beta_4^2)
\end{aligned} \tag{B.56}$$

Generalización caso perdedor.

$$\begin{aligned}
\hat{m}_{f_{t_b} \rightarrow p_3}(p_3) &= \int \int \mathbb{I}(t_b = p_3 + \dots + P_N) \mathcal{N}(t_b | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_4 | \mu_4, \sigma_4^2 + \beta_4^2) \\
&\quad \mathcal{N}(p_5 | \mu_5, \sigma_5^2 + \beta_5^2) \dots \mathcal{N}(p_N | \mu_N, \sigma_N^2 + \beta_N^2) dt_b dp_4 dp_5 \dots dp_N \\
&= \mathcal{N}(p_3 | (\mu_a - \sum_{i \in A_b, i \neq 3} \mu_i) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \underbrace{\sum_{i \in A_b, i \neq 3} (\sigma_i^2 + \beta_i^2)}_{\sigma_b^2 - \sigma_3^2 - \beta_3^2}) \\
&= \mathcal{N}(p_3 | (\mu_a - (\mu_b - \mu_3)) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_b^2 - (\sigma_3^2 + \beta_3^2)) \\
&= \mathcal{N}(p_3 | (\delta + \mu_3) - \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2 - \beta_3^2))
\end{aligned} \tag{B.57}$$

Donde en este caso, todos los jugadores son del mismo equipo t_b

$\hat{m}_{p_1 \rightarrow f_{p_1}}(p_1)$ (Caso ganador)

$$\begin{aligned}
m_{p_1 \rightarrow f_{p_1}}(p_1) &= \prod_{g \in n(p_1) \setminus \{f_{p_1}\}} m_{g \rightarrow p_1}(p_1) = m_{f_{t_a} \rightarrow p_1}(p_1) \\
&\approx \hat{m}_{f_{t_a} \rightarrow p_1}(p_1) = \mathcal{N}(p_1 | (\mu_b - \mu_2) + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2 + \beta_2^2)
\end{aligned} \tag{B.58}$$

$\hat{m}_{p_3 \rightarrow f_{p_3}}(p_3)$ (Caso perdedor)

$$\begin{aligned}
m_{p_3 \rightarrow f_{p_3}}(p_3) &= \prod_{g \in n(p_3) \setminus \{f_{p_3}\}} m_{g \rightarrow p_3}(p_3) = m_{f_{t_b} \rightarrow p_3}(p_3) \\
&\approx \hat{m}_{f_{t_b} \rightarrow p_3}(p_3) = \mathcal{N}(p_3 | (\mu_a - \mu_4) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2 + \beta_4^2)
\end{aligned} \tag{B.59}$$

$\hat{m}_{f_{p_1} \rightarrow s_1}(s_1) :$ (Caso ganador)

$$\begin{aligned}
m_{f_{p_1} \rightarrow s_1}(s_1) &= \int \cdots \int f_{p_1}(\mathbf{x}) \prod_{h \in n(f_{p_1}) \setminus \{s_1\}} m_{h \rightarrow f_{p_1}}(h) d\mathbf{x}_{\setminus \{s_1\}} \\
&= \int \cdots \int f_{p_1}(p_1) m_{p_1 \rightarrow f_{p_1}}(p_1) d\mathbf{x}_{\setminus \{s_1\}} \\
&\approx \int \cdots \int f_{p_1}(p_1) \hat{m}_{p_1 \rightarrow f_{p_1}}(p_1) d\mathbf{x}_{\setminus \{s_1\}} \tag{B.60} \\
&= \int \mathcal{N}(p_1 | s_1, \beta_1^2) \mathcal{N}(p_1 | -\delta + \mu_1 + \delta_{\div}, \vartheta_{\div}^2 + \vartheta^2 - \sigma_1^2 - \beta_1^2) dp_1 \\
&= \mathcal{N}(s_1 | (\mu_b - \mu_2) + \delta_{\div}, \beta_1^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2 + \beta^2) \\
&= \mathcal{N}(s_1 | (\mu_b - \mu_2) + \delta_{\div}, \beta_1^2 + \beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2)
\end{aligned}$$

Generalización caso ganador

$$\begin{aligned}
m_{f_{p_1} \rightarrow s_1}(s_1) &= \int \cdots \int f_{p_1}(\mathbf{x}) \prod_{h \in n(f_{p_1}) \setminus \{s_1\}} m_{h \rightarrow f_{p_1}}(h) d\mathbf{x}_{\setminus \{s_1\}} \\
&= \int \cdots \int f_{p_1}(p_1) m_{p_1 \rightarrow f_{p_1}}(p_1) d\mathbf{x}_{\setminus \{s_1\}} \\
&\approx \int \cdots \int f_{p_1}(p_1) \hat{m}_{p_1 \rightarrow f_{p_1}}(p_1) d\mathbf{x}_{\setminus \{s_1\}} \tag{B.61} \\
&= \int \mathcal{N}(p_1 | s_1, \beta_1^2) \mathcal{N}(p_1 | (-\delta + \mu_1) + \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2 - \beta_1^2)) dp_1 \\
&= \mathcal{N}(s_1 | (-\delta + \mu_1) + \delta_{\div}, \beta_1^2 + \vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2 - \beta_1^2)) \\
&= \mathcal{N}(s_1 | (-\delta + \mu_1) + \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2))
\end{aligned}$$

$\hat{m}_{f_{p_3} \rightarrow s_3}(s_3) :$ (Caso perdedor)

$$\begin{aligned}
m_{f_{p_3} \rightarrow s_3}(s_3) &= \int \cdots \int f_{p_3}(\mathbf{x}) \prod_{h \in n(f_{p_3}) \setminus \{s_3\}} m_{h \rightarrow f_{p_3}}(h) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \cdots \int f_{p_3}(p_3) m_{p_3 \rightarrow f_{p_3}}(p_3) d\mathbf{x}_{\setminus \{s_3\}} \\
&\approx \int \cdots \int f_{p_3}(p_3) \hat{m}_{p_3 \rightarrow f_{p_3}}(p_3) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \cdots \int \mathcal{N}(p_3 | s_3, \beta^2) \mathcal{N}(p_3 | (\mu_b - \mu_4) + \delta_{\div}, \vartheta_{\div}^2 + \sigma_b^2 + \sigma_4^2 + \beta^2) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \mathcal{N}(p_3 | s_3, \beta^2) \mathcal{N}(p_3 | (\mu_a - \mu_4) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2 + \beta^2) dp_3 \\
&= \mathcal{N}(s_3 | (\mu_a - \mu_4) - \delta_{\div}, \beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2 + \beta^2) \\
&= \mathcal{N}(s_3 | (\mu_a - \mu_4) - \delta_{\div}, 2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2) \tag{B.62}
\end{aligned}$$

Generalización caso perdedor

$$\begin{aligned}
m_{f_{p_3} \rightarrow s_3}(s_3) &= \int \cdots \int f_{p_3}(\mathbf{x}) \prod_{h \in n(f_{p_3}) \setminus \{s_3\}} m_{h \rightarrow f_{p_3}}(h) d\mathbf{x}_{\setminus \{s_3\}} \\
&= \int \cdots \int f_{p_3}(p_3) m_{p_3 \rightarrow f_{p_3}}(p_3) d\mathbf{x}_{\setminus \{s_3\}} \\
&\approx \int \cdots \int f_{p_3}(p_3) \hat{m}_{p_3 \rightarrow f_{p_3}}(p_3) d\mathbf{x}_{\setminus \{s_3\}} \tag{B.63} \\
&= \int \mathcal{N}(p_3 | s_3, \beta_3^2) \mathcal{N}(p_2 | (\delta + \mu_3) - \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2 - \beta_3^2)) dp_3 \\
&= \mathcal{N}(s_3 | (\delta + \mu_3) - \delta_{\div}, \beta_3^2 + \vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2 - \beta_3^2)) \\
&= \mathcal{N}(s_3 | (\delta + \mu_3) - \delta_{\div}, \vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2))
\end{aligned}$$

Ganador

$$\begin{aligned}
p(s_1 | o, A) &= \prod_{h \in n(x_i)} m_{h \rightarrow x_i} = m_{f_{s_1} \rightarrow s_1}(s_1) m_{f_{p_1} \rightarrow s_1}(s_1) \\
&= \mathcal{N}(s_1 | \mu_1, \sigma_1^2) \mathcal{N}(s_1 | (\mu_b - \mu_2) + \delta_{\div}, 2\beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2) \\
&= \mathcal{N}\left(s_1 \left| \frac{\mu_1(2\beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2) + [(\mu_b - \mu_2) + \delta_{\div}]\sigma_1^2}{\sigma_1^2 + [2\beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2]}, \frac{\sigma_1^2[2\beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2]}{\sigma_1^2 + [2\beta^2 + \vartheta_{\div}^2 + \sigma_b^2 + \sigma_2^2]} \right.\right) \tag{B.64}
\end{aligned}$$

Generalización ganador

$$\begin{aligned}
p(s_1 | o, A) &= \prod_{h \in n(x_i)} m_{h \rightarrow x_i} = m_{f_{s_1} \rightarrow s_1}(s_1) m_{f_{p_1} \rightarrow s_1}(s_1) \\
&= \mathcal{N}(s_1 | \mu_1, \sigma_1^2) \mathcal{N}(s_1 | \underbrace{(-\delta + \mu_1) + \delta_{\div}}_{=\mu}, \underbrace{\vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2)}_{=\sigma^2}) \\
&= \mathcal{N}\left(s_1 \left| \frac{(\mu_1 \sigma^2 + \mu \sigma_1^2)}{(\sigma_1^2 + \sigma^2)}, \frac{(\sigma_1^2 \sigma^2)}{(\sigma_1^2 + \sigma^2)} \right.\right) \\
&= \mathcal{N}\left(s_1 \left| \frac{\mu_1[\vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2)] + [(-\delta + \mu_1) + \delta_{\div}]\sigma_1^2}{(\sigma_1^2 + [\vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2)])}, \frac{(\sigma_1^2[\vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2)])}{(\sigma_1^2 + [\vartheta_{\div}^2 + (\vartheta^2 - \sigma_1^2)])} \right.\right) \\
&= \mathcal{N}\left(s_1 \left| \frac{\mu_1(\vartheta_{\div}^2 + \vartheta^2) + \sigma_1^2(\delta_{\div} - \delta)}{(\vartheta_{\div}^2 + \vartheta^2)}, \frac{\sigma_1^2(\vartheta_{\div}^2 + \vartheta^2) - \sigma_1^4}{(\vartheta_{\div}^2 + \vartheta^2)} \right.\right) \\
&= \mathcal{N}\left(s_1 \left| \frac{\mu_1(\vartheta_{\div}^2 + \vartheta^2) + \sigma_1^2(\delta_{\div} - \delta)}{(\vartheta_{\div}^2 + \vartheta^2)}, \frac{\sigma_1^2(\vartheta_{\div}^2 + \vartheta^2) - \sigma_1^4}{(\vartheta_{\div}^2 + \vartheta^2)} \right.\right) \tag{B.65}
\end{aligned}$$

Perdedor

$$\begin{aligned}
p(s_3|o, A) &= \prod_{h \in n(x_i)} m_{h \rightarrow x_i} = m_{f_{s_3} \rightarrow s_3}(s_3) m_{f_{p_3} \rightarrow s_3}(s_3) \\
&= \mathcal{N}(s_3|\mu_3, \sigma_3^2) \mathcal{N}(s_3|(\mu_a - \mu_4) - \delta_{\div}, 2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2) \\
&= \mathcal{N}\left(s_3 \left| \frac{\mu_3[2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2] + [(\mu_a - \mu_4) - \delta_{\div}]\sigma_3^2}{\sigma_3^2 + (2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)}, \frac{\sigma_3^2(2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)}{\sigma_3^2 + (2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)} \right| \right) \\
&= \mathcal{N}\left(s_3 \left| \frac{\mu_3[2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2] + [(\mu_a - \mu_4) - \delta_{\div}]\sigma_3^2}{\sigma_3^2 + (2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)}, \frac{\sigma_3^2(2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)}{\sigma_3^2 + (2\beta^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_4^2)} \right| \right)
\end{aligned} \tag{B.66}$$

Generalización perdedor

$$\begin{aligned}
p(s_3|o, A) &= \prod_{h \in n(x_i)} m_{h \rightarrow x_i} = m_{f_{s_3} \rightarrow s_3}(s_3) m_{f_{p_3} \rightarrow s_3}(s_3) \\
&= \mathcal{N}(s_3|\mu_3, \sigma_3^2) \mathcal{N}(s_3| \underbrace{(\delta + \mu_3) - \delta_{\div}}_{=\mu}, \underbrace{\vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2)}_{=\sigma^2}) \\
&= \mathcal{N}\left(s_3 \left| \frac{(\mu_3\sigma^2 + \mu\sigma_3^2)}{(\sigma_3^2 + \sigma^2)}, \frac{(\sigma_3^2\sigma^2)}{(\sigma_3^2 + \sigma^2)} \right| \right) \\
&= \mathcal{N}\left(s_3 \left| \frac{(\mu_3[\vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2)] + [(\delta + \mu_3) - \delta_{\div}]\sigma_3^2)}{(\sigma_3^2 + [\vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2)])}, \frac{(\sigma_3^2[\vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2)])}{(\sigma_3^2 + [\vartheta_{\div}^2 + (\vartheta^2 - \sigma_3^2)])} \right| \right) \\
&= \mathcal{N}\left(s_3 \left| \frac{\mu_3(\vartheta_{\div}^2 + \vartheta^2) + \sigma_3^2(\delta - \delta_{\div})}{(\vartheta_{\div}^2 + \vartheta^2)}, \frac{\sigma_3^2(\vartheta_{\div}^2 + \vartheta^2) - \sigma_3^4}{(\vartheta_{\div}^2 + \vartheta^2)} \right| \right)
\end{aligned} \tag{B.67}$$

B.2 Ejemplo 1 vs 1-Handicap

Para el caso de esta Tesis, se hará brevemente las cuentas y mensajes para **TrueSkill** para el ejemplo de un evento entre dos jugadores, en el cual uno de ellos (el jugador de ficha negra), tiene *handicap* a favor. Este ejemplo es el caso típico a lo largo de este trabajo. Una de las hipótesis que se explica en la sección de resultados 4.2, es modelar a las fichas de handicap como un jugador mas. Este juega en un equipo con el jugador negro. Muchas de las cuentas intermedias se encuentran en el apéndice B.1.1 y B.1.2.

El grafo para este caso es el que se puede observar en la figura B.1.

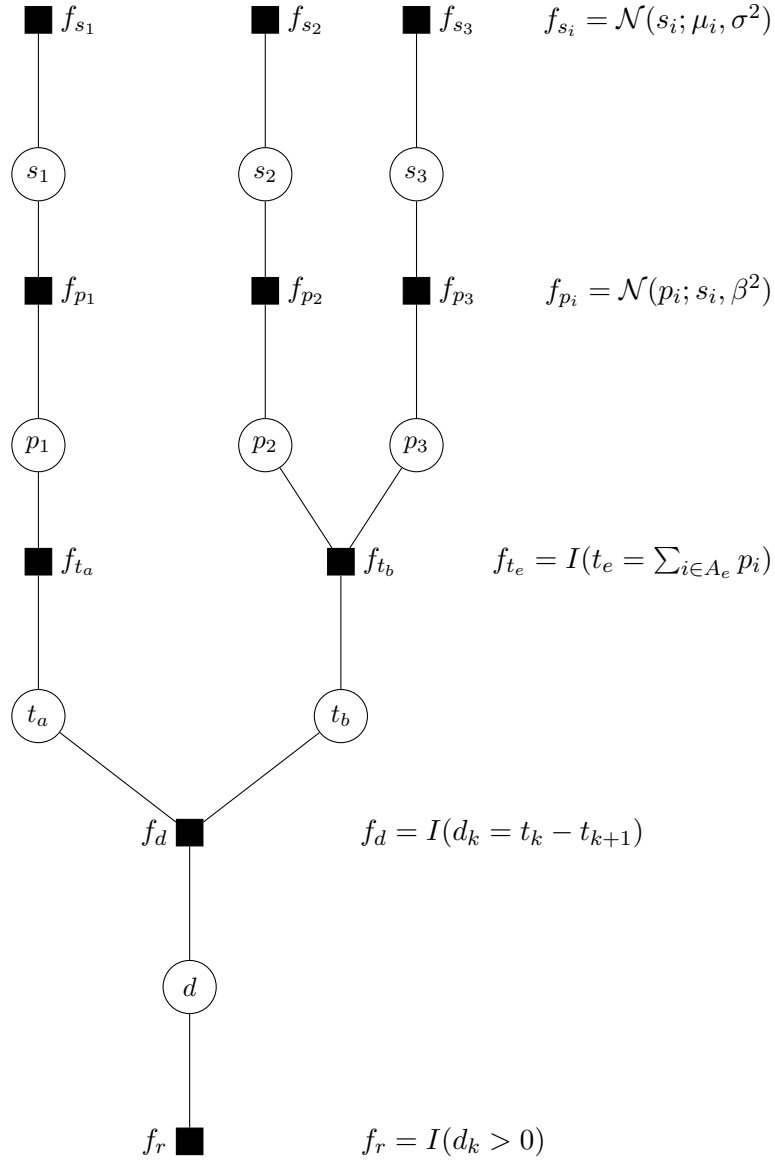


Figura B.1: Grafo bipartito de la factorización (*factor graph*) del modelo **Trueskill** para el caso particular de un jugador enfrentando a otro en el cual cuenta con ayuda *dehandicap*.

Veamos el caso del cálculo de la *posterior* del jugador negro. Para ello realizamos la inferencia, calculando la marginal del jugador negro (s_2). Los mensajes para este caso se pueden ver la figura B.2

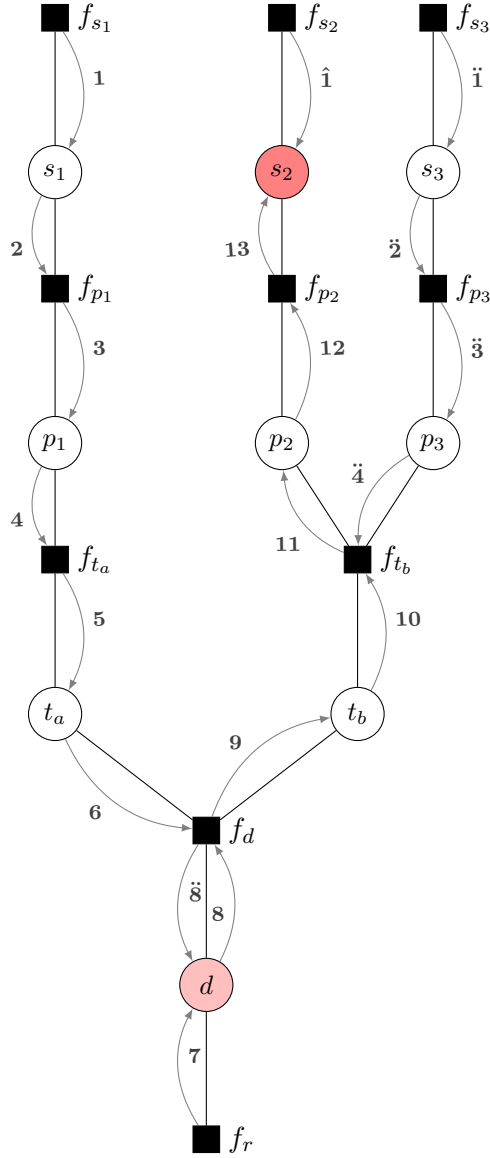


Figura B.2: Grafo bipartito de la factorización (*factor graph*) del modelo **Trueskill** para el caso particular de un jugador enfrentando a otro en el cual cuenta con ayuda *dehandicap*. Se muestran los mensajes para el cálculo del *posterior* del jugador negro.

El mensaje (1) (como el $\hat{1}$ y \hat{i}) devuelve el factor gaussiana mismo, el mensaje (2) (y $\tilde{2}$) es solamente el producto de todos los mensajes de llegada en el borde del nodo s_1 (s_3) y como solo hay un mensaje de llegada, solamente copiamos ese como mensaje de salida.

$$m_{s_i \rightarrow f_{p_i}}(s_i) = \mathcal{N}(s_i | \mu_i, \sigma_i^2) \quad (\text{B.68})$$

Luego tenemos que computar el mensaje (3) (y $\tilde{3}$). El algoritmo de *belief propagation* nos dice que tenemos que multiplicar el mensaje de entrada (2) (y el $\tilde{2}$) por el factor gaussiana y luego integrar sobre la variable s_2 (y s_3 respectivamente). Estos mensajes son una gaussiana donde su media es la media de la distribución de f_{s_2} (f_{s_3}) y que su varianza es la suma de la varianza de f_{s_2} y f_{p_2} (f_{s_3} y f_{p_3}). Este proceso de solapar una gaussiana usando otra gaussiana es un ejemplo de la operación matemática denominada **convolución**.

$$m_{f_{p_i} \rightarrow p_i}(p_i) = \int \cdots \int f_{p_i}(p_i) m_{s_i \rightarrow f_{p_i}}(s_i) d\mathbf{x}_{\setminus \{p_i\}} = \mathcal{N}(p_i | \mu_i, \beta_i^2 + \sigma_i^2) \quad (\text{B.69})$$

El mensaje (4) (y $\ddot{4}$), es simplemente la copia del mensaje (3) ($\ddot{3}$ respectivamente) ya que es el único mensaje entrante del nodo p_2 (p_3).

Luego de acá, hay que ver el sentido de cada mensaje individual y veremos que para poder enviar el mensaje del nodo de equipos del equipo b (f_{t_b}), se requiere calcular el resto de los mensajes. Por un lado tenemos el mensaje (5), el cual como tenemos un único integrante, este mensaje como el (6), son una copia del mensaje (4). Si suponemos que el jugador blanco es el ganador, r_{ab} es verdadero, equipo a gana, el mensaje (7) es una indicadora con $d_{ab} > 0$. Para poder calcular el mensaje (8), debemos primero calcular la marginal del nodo d y realizar la aproximación. Esto es la multiplicación de los mensajes entrantes. Este mensaje es igual a la ecuación 2.32

Para computar el mensaje (9), tomamos el factor f_d , multiplicado por los dos mensajes entrantes (6) y (8), y luego se suma sobre r_{ab} y se integra sobre t_a . Debido a que estamos considerando que el jugador blanco es el ganador (equipo a), el mensaje (8) evalúa a 1 a este caso, y 0 en caso contrario.

$$\begin{aligned} \hat{m}_{f_d \rightarrow t_b}(t_b) &= \int \int \mathbb{I}(d = t_a - t_b) \mathcal{N}(d | \delta_{\div}, \vartheta_{\div}^2) \mathcal{N}(t_a | \mu_a, \sigma_a^2) dd dt_a \\ &= \mathcal{N}(t_b | \mu_a + \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \end{aligned} \quad (\text{B.70})$$

Mensaje (10) es una copia del mensaje (9) ya que es el único mensaje de entrada para el nodo f_{t_b} . El mensaje (11), se calcula con la multiplicación de mensajes entrantes, sin contar el que podría llegar del nodo p_2 e integrando sobre todas las variables menos esta.

$$\begin{aligned} \hat{m}_{f_{t_b} \rightarrow p_2}(p_2) &= \int \int \mathbb{I}(t_b = p_2 + p_3) \mathcal{N}(t_b | \mu_a - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2) \mathcal{N}(p_3 | \mu_3, \sigma_3^2 + \beta_3^2) dt_b dp_3 \\ &= \mathcal{N}(p_2 | (\mu_a - \mu_3) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_3^2 + \beta_3^2) \end{aligned} \quad (\text{B.71})$$

El mensaje (12) es una copia del (11). El mensaje (13) se computa multiplicando el factor gaussiana, describiendo la variabilidad del rendimiento, con el mensaje de llegada (12) y luego integrando sobre p_2 . Esto es de nuevo una convolución.

$$\begin{aligned} \hat{m}_{f_{p_2} \rightarrow s_2}(s_2) &= \int \mathcal{N}(p_2 | s_2, \beta_2^2) \mathcal{N}(p_2 | (\mu_a - \mu_3) - \delta_{\div}, \vartheta_{\div}^2 + \sigma_a^2 + \sigma_3^2 + \beta_3^2) dp_2 \\ &= \mathcal{N}(s_2 | (\mu_a - \mu_3) - \delta_{\div}, \beta_2^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_3^2 + \beta_3^2) \end{aligned} \quad (\text{B.72})$$

Finalmente el mensaje ($\hat{1}$) es la distribución gaussiana de la habilidad a priori del jugador negro.

Para obtener la distribución marginal de s_2 , se multiplica el mensaje (13) y ($\hat{1}$). Obteniendo de esta manera la *posterior* del jugador negro, que al haber utilizado la aproximación su distribución de probabilidad sigue siendo una gaussiana.

$$\begin{aligned}
p(s_2) &= \mathcal{N}(s_2 | \mu_2, \sigma_2^2) \mathcal{N}(s_2 | \underbrace{(\mu_a - \mu_3) - \delta_{\div}}_{=\mu}, \underbrace{\beta_2^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_3^2 + \beta_3^2}_{=\sigma^2}) \\
&= \mathcal{N}(s_2 | (\mu_a - \mu_3) + \delta_{\div}, \beta_2^2 + \vartheta_{\div}^2 + \sigma_a^2 + \sigma_3^2 + \beta_3^2) \\
&= \mathcal{N}\left(s_2 | \frac{(\mu_2 \sigma^2 + \mu \sigma_2^2)}{(\sigma_2^2 + \sigma^2)}, \frac{(\sigma_2^2 \sigma^2)}{(\sigma_2^2 + \sigma^2)}\right)
\end{aligned} \tag{B.73}$$

De esta forma obtuvimos la actualización de habilidad del jugador negro. De forma análoga se realiza la marginalización para el nodo de habilidad del jugador blanco y el de *handicap*.

B.2.1 Full TrueSkill

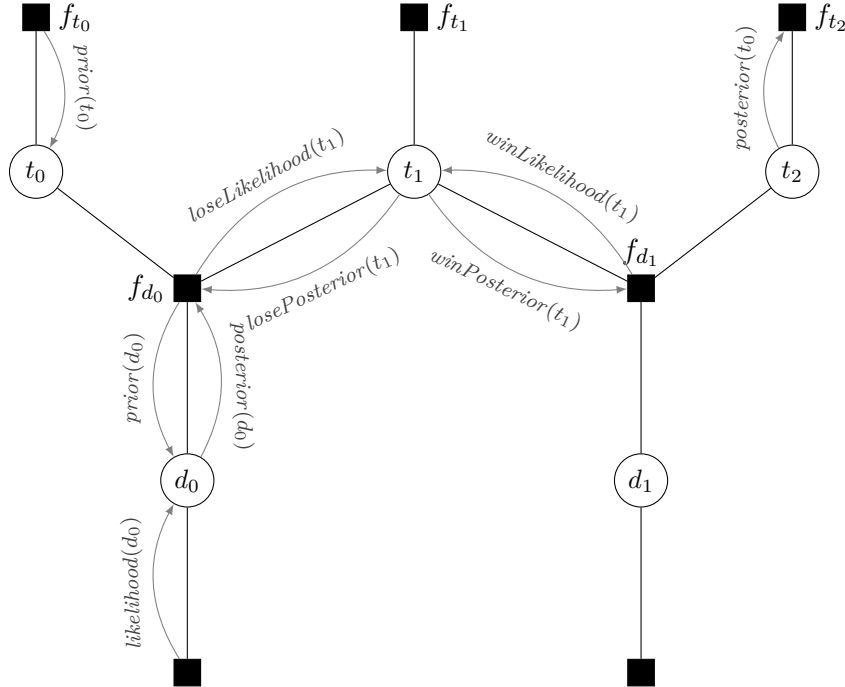


Figura B.3: Grafo bipartito de la factorización (*factor graph*) del modelo TrueSkill para el caso de multi-equipos. Se recortaron algunos nodos para una mejor visualización de los cambios relevantes al caso de TrueSkill explicado anteriormente.

Sea la marginal $p(d_i)$,

$$p(d_i) = \text{prior}(d_i) \cdot \text{likelihood}(d_i) \tag{B.74}$$

winPosterior(t_i): $m_{t_i \rightarrow f_{d_i}}(t_i)$

$$\text{winPosterior}(t_i) = \text{prior}(t_i) \cdot \text{winLikelihood}(t_i) \tag{B.75}$$

losePosterior(t_i): $m_{t_i \rightarrow f_{d_{i-1}}}(t_i)$ Se inician con una forma neutra

$$\text{losePosterior}(t_i) = \text{prior}(t_i) \cdot \text{loseLikelihood}(t_i) \tag{B.76}$$

prior(d_i): $m_{f_{d_0} \rightarrow d_0}(d_0)$

$$prior(d_i) = \iint \mathbb{I}(d_i = t_i - t_{i+1}) \cdot winPosterior(t_i) \cdot losePosterior(t_{i+1}) dt_i dt_{i+1} \quad (B.77)$$

$\hat{p}(d_i)$:

$$\hat{p}(d_i) = prior(d_i) \widehat{likelihood}(d_i) \quad (B.78)$$

Siguiendo el algoritmo de *expectation propagation*, aproximamos el $likelihood(d_i)$, mediante la aproximacion de la marginal $p(d_i)$ mediante *moment matching*, resultando en $\hat{p}(d_i)$ que es una gaussiana con misma media y varianza que $p(d_i)$.

$$\hat{p}(d_i) = approx(p(d_i)) = approx(prior(d_i) \cdot likelihood(d_i)) \quad (B.79)$$

Entonces,

$\widehat{likelihood}(d_i)$:

$$\widehat{likelihood}(d_i) = \frac{approx(prior(d_i) \cdot likelihood(d_i))}{prior(d_i)} \quad (B.80)$$

$\widehat{posterior}(d_i)$:

$$\widehat{posterior}(d_i) = \widehat{likelihood}(d_i) \quad (B.81)$$

loseLikelihood(t_{i+1}): $m_{f_{d_i} \rightarrow t_{i+1}}(t_{i+1})$

$$loseLikelihood(t_{i+1}) = \iint \mathbb{I}(d_i = t_i - t_{i+1}) winPosterior(t_i) \widehat{posterior}(d_i) dd_i dt_{i+1} \quad (B.82)$$

winLikelihood(t_i): $m_{f_{d_{i-1}} \rightarrow t_i}(t_i)$ Se inicializan con su forma neutra

$$winLikelihood(t_i) = \iint \mathbb{I}(d_i = t_i - t_{i+1}) losePosterior(t_{i+1}) \widehat{posterior}(d_i) dd_i dt_{i+1} \quad (B.83)$$

B.2.2 TrueSkill Through Time

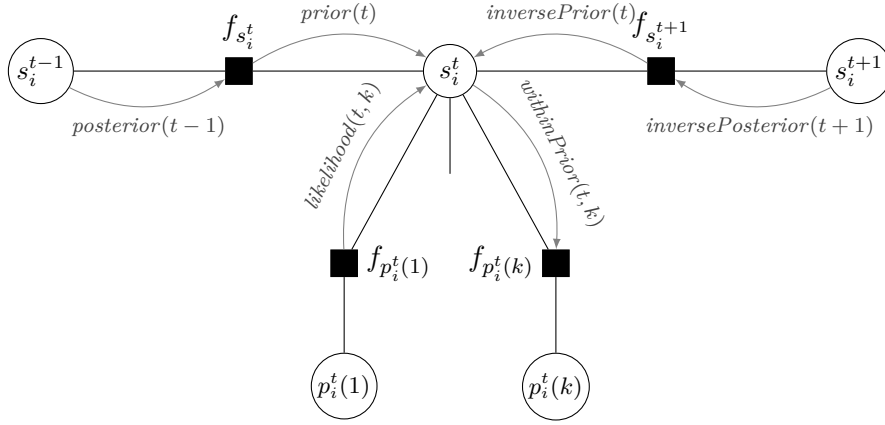


Figura B.4: Representación del grafo bipartito de la factorización (*factor graph*) del modelo TTT para el caso de la habilidad de un jugador i .

$posterior_i(t-1)$:

$$\begin{aligned}
 posterior_i(t-1) &= m_{s_i^{t-1} \rightarrow f_{s_i^t}}(s_i^{t-1}) \\
 &= prior_i(t-1) \cdot \prod_{k=1}^{K_i^{t-1}} likelihood_i(t-1, k) \\
 &= p(s_i^{t-1}) inversePrior_i(t-1)^{-1}
 \end{aligned} \tag{B.84}$$

$prior_i(t)$:

$$\begin{aligned}
 prior_i(t) &= m_{f_{s_i^t} \rightarrow s_i^t}(s_i^t) \\
 &= \int f_{s_i^t} \cdot posterior_i(t-1) ds_i^{t-1}
 \end{aligned} \tag{B.85}$$

$likelihood_i(t, k)$:

$$\begin{aligned}
 likelihood_i(t, k) &= m_{f_{p_i^t(k)} \rightarrow s_i^t}(s_i^t) \\
 &= trueSkillVariationalLikelihood_i(t, k)
 \end{aligned} \tag{B.86}$$

$withinPrior_i(t, k)$:

$$\begin{aligned}
 withinPrior_i(t, k) &= m_{s_i^t \rightarrow f_{p_i^t(k)}}(s_i^t) \\
 &= prior_i(t) \cdot inversePrior_i(t) \cdot \prod_{q=1, q \neq k}^{K_i^t} likelihood_i(t, q) \\
 &= p(s_i^t) \cdot likelihood_i(t, k)^{-1}
 \end{aligned} \tag{B.87}$$

inversePrior_i(t):

$$\begin{aligned}
inversePrior_i(t) &= m_{f_{s_i^{t+1} \rightarrow s_i^t}}(s_i^t) \\
&= \int f_{s_i^{t+1}} \cdot inversePosterior_i(t+1) ds_i^{t+1}
\end{aligned} \tag{B.88}$$

inversePosterior_i(t+1):

$$\begin{aligned}
inversePosterior_i(t+1) &= m_{s_i^{t+1} \rightarrow f_{s_i^{t+1}}}(s_i^{t+1}) \\
&= inversePrior_i(t+1) \cdot \prod_{k=1}^{K_i^{t+1}} likelihood_i(t+1, k) \\
&= p(s_i^{t+1}) \cdot prior_i(t+1)^{-1}
\end{aligned} \tag{B.89}$$