Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Mathe III

Ralf Herbrich

Bayesianische Statistik

Overview

1. Bayesian Probabilities and Conjugacy
2. Graphical Models
3. The Sum-Product Algorithm
4. Bayesian Ranking: TrueSkill
5. Information Theory

Overview

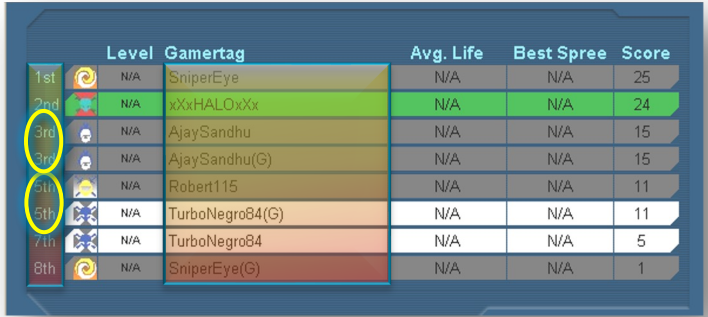
1. Bayesian Probabilities and Conjugacy
2. Graphical Models
3. The Sum-Product Algorithm
4. **Bayesian Ranking: TrueSkill**
5. Information Theory

The Skill Rating Problem

- **Given**:
  - **Match outcomes**: Orderings among k teams consisting of $n_1, n_2, \dots, n_k$ players.

Team		Score
1st	Red Team	50
2nd	Blue Team	40

	Level	Gamertag	Avg. Life	Best Spree	Score
1st	10	BlueBot	00:00:49	6	15
1st	7	SniperEye	00:00:41	4	14
1st	9	ProThepirate	00:01:07	3	13
1st	10	dazdemon	00:00:59	3	8
2nd	10	WastedHarry	00:00:41	4	17
2nd	3	Ascla	00:00:37	2	10
2nd	9	Antidote4Losing	00:00:41	2	9
2nd	12	Blackknight9	00:00:48	3	4



	Level	Gamertag	Avg. Life	Best Spree	Score
1st	N/A	SniperEye	N/A	N/A	25
2nd	N/A	xXxHALOxXx	N/A	N/A	24
3rd	N/A	AjaySandhu	N/A	N/A	15
3rd	N/A	AjaySandhu(G)	N/A	N/A	15
5th	N/A	Robert115	N/A	N/A	11
5th	N/A	TurboNegro84(G)	N/A	N/A	11
7th	N/A	TurboNegro84	N/A	N/A	5
8th	N/A	SniperEye(G)	N/A	N/A	1

- **Questions**:
  1. Skill s_i for each player such that $s_i > s_j \Leftrightarrow P(\text{Player } i \text{ wins}) > P(\text{Player } j \text{ wins})$
  2. Global ranking among all players
  3. Fair matches between teams of players

Two-Player Match Outcome Model

- **Simple Two-Player Games**: Our data is the identity i and j of the two players and the outcome $y \in \{-1, +1\}$ of a match between them
  - **Bradley-Terry Model (1952)**: Model of a win given skills s_i and s_j is

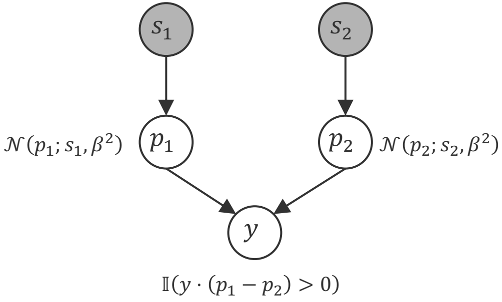
$$P\left(y = 1 | s_i, s_j\right) = \frac{\exp(s_i)}{\exp(s_i) + \exp\left(s_j\right)} = \boxed{\frac{\exp\left(s_i - s_j\right)}{1 + \exp\left(s_i - s_j\right)}}$$

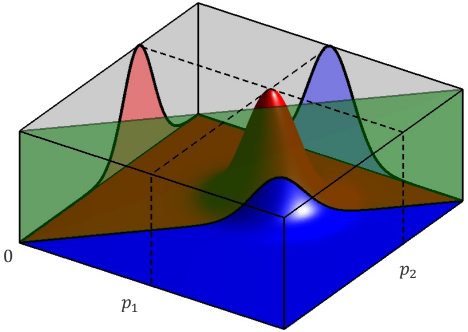
Logistic sigmoid in skill difference

  - **Thurstone Case V Model (1927)**: Model of a win given skills s_i and s_j is

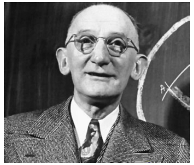
$$P\left(y = 1 | s_i, s_j\right) = \boxed{\int_0^\infty \mathcal{N}\left(t; s_i - s_j, 2\beta^2\right) dt}$$

Probit sigmoid in skill difference





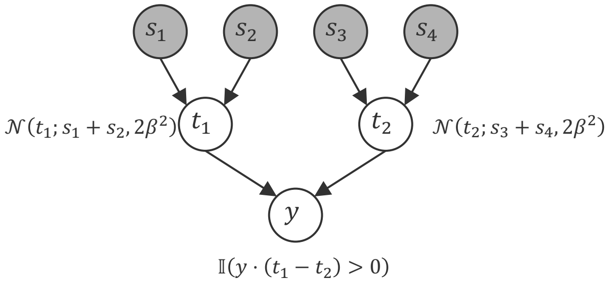
Ralph A. Bradley (1923 – 2001)



Louis Leon Thurstone (1887 – 1955)

Two-Team Match Outcome Model

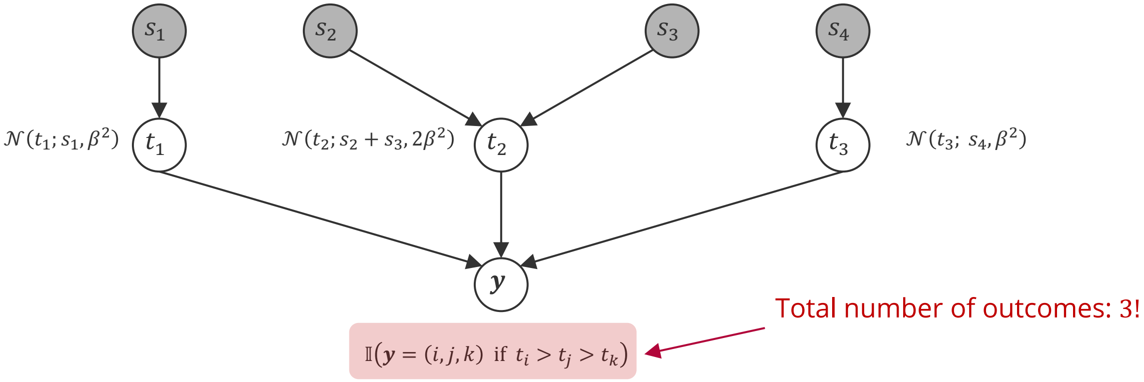
- **Team Assumption**: Skill of a team is the sum of the skill of its players



- **Pro**: Games where the team scores are additive (e.g., kill count in first-person shooter)
- **Con**: Games where the outcome is determined by a single player (e.g., fastest car in a race)
- **Observation**: Match outcomes correlate the skills of players
  - **Same Team**: Anti-correlated
  - **Opposite Teams**: Correlated

Multi-Team Match Outcome Model

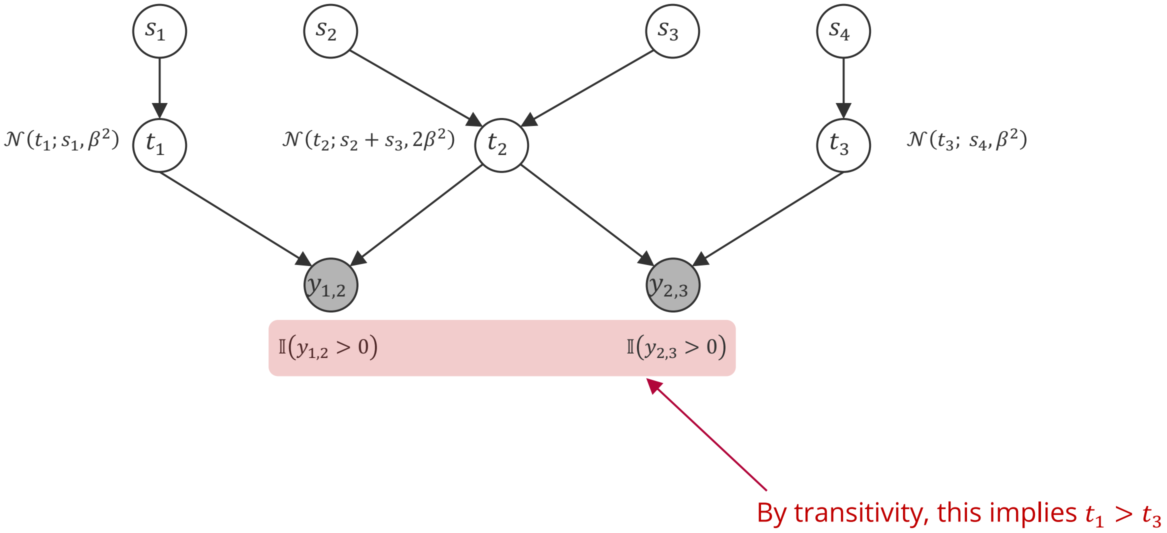
- **Possible Outcomes**: Permutations $\boldsymbol{y} \in \{1,2,3\}^3$ of players



- **Easy** to **sample** for given skills but computationally difficult to "invert"!

From Match Outcomes to Pairwise Rankings

- **Learning**: In the ranking setting, we observe multi-team match outcomes and want to infer the skills!
- **Idea**: Leverage the transitivity of the real line of latent scores!



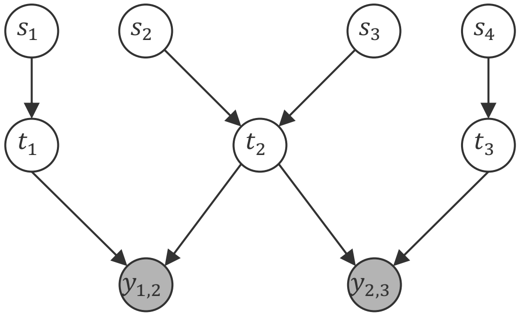
Mathe III

Unit 13b – Bayesianische Statistik

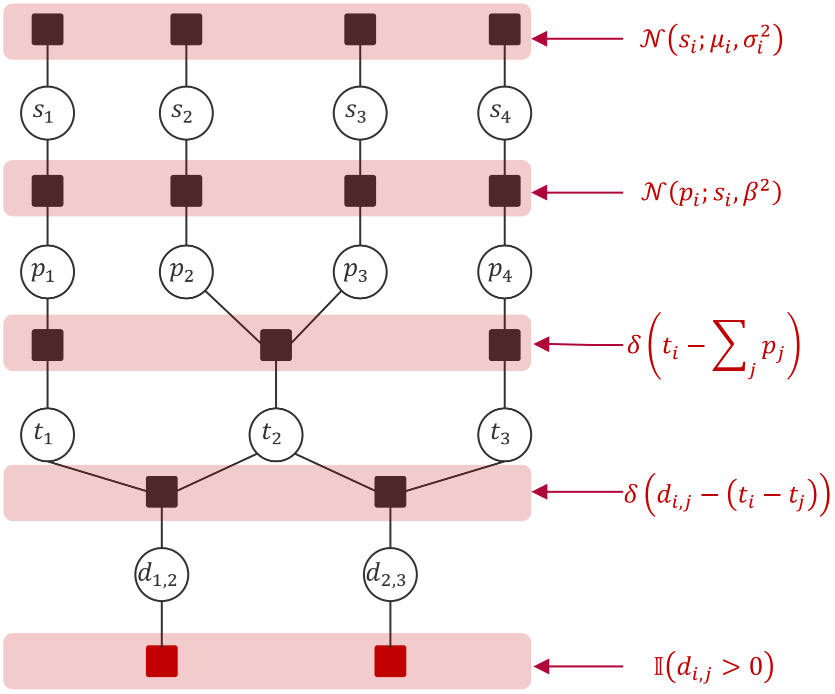
TrueSkill Factor Graphs

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Bayesian Network

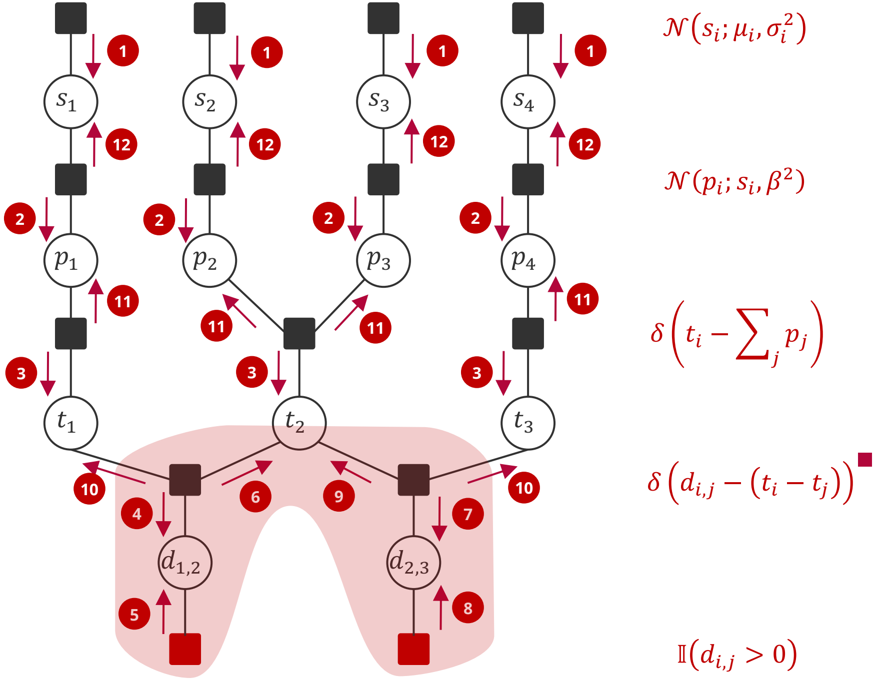


Factor Graph



(Approximate) Message Passing in TrueSkill Factor Graphs

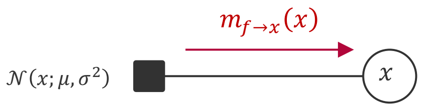
TrueSkill Factor Graph



- Four Phases
  1. Pass prior messages (1)
  2. Pass messages *down* to the team performances (2 to 3)
  3. Iterate the approximate messages on the pairwise team differences (4 to 9)
  4. Pass messages back from *up* from team performances to player skill (10 – 12)
- Since this is a *tree,* the algorithm is guaranteed to converge!

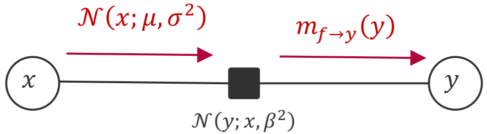
Message Update Equations

Gaussian Factor



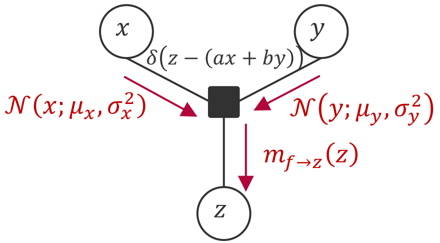
$$m_{f\to x}(x) = \mathcal{N}(x;\mu,\sigma^2)$$

Gaussian Mean Factor



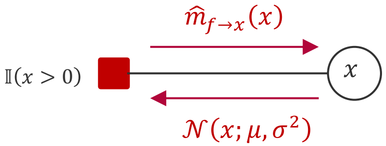
$$m_{f\to y}(y) = \int \mathcal{N}(y;x,\beta^2)\cdot\mathcal{N}(x;\mu,\sigma^2)\,dx = \mathcal{N}(y;\mu,\sigma^2+\beta^2)$$

Weighted Sum Factor



$$m_{f\to z}(z) = \mathcal{N}\left(z;a\mu_x+b\mu_y,a^2\sigma_x^2+b^2\sigma_y^2\right)$$

Greater-Than Factor



$$\widehat{m}_{f\to x}(x) = \frac{\hat{p}(x)}{m_{x\to f}(x)} = \frac{\mathcal{N}(x;\hat{\mu},\hat{\sigma}^2)}{\mathcal{N}(x;\mu,\sigma^2)}$$

Mean and variance of a truncated Gaussian $\mathcal{N}(x;\mu,\sigma^2)$

Truncated Gaussians

- **Truncated Gaussians**. *A truncated Gaussian given by $p(x) \propto \mathbb{I}(x > 0) \cdot \mathcal{N}(x; \mu, \sigma^2)$ has the following three moments*

$$Z(\mu,\sigma) = \int_{-\infty}^{+\infty} p(x)\,dx = 1 - F(0;\mu,\sigma^2)$$

Follows from definition of F

$$E[X] = \int_{-\infty}^{+\infty} x \cdot p(x)\,dx = \mu + \sigma \cdot v\left(\frac{\mu}{\sigma}\right)$$

Additive update that goes to zero as $\frac{\mu}{\sigma} \to \infty$

$$\mathrm{var}[X] = \int_{-\infty}^{+\infty} (x - E[X])^2 \cdot p(x)\,dx = \sigma^2 \cdot \left(1 - w\left(\frac{\mu}{\sigma}\right)\right)$$

Multiplicative update that goes to 1 as $\frac{\mu}{\sigma} \to \infty$

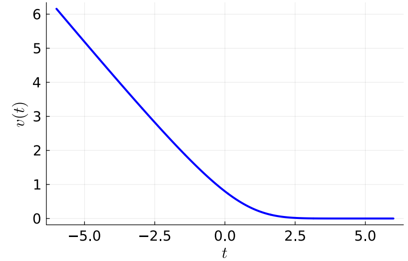
where the probit $F(t;\mu,\sigma^2) \coloneqq \int_{-\infty}^{t} \mathcal{N}(x;\mu,\sigma^2)\,dx$ *and*

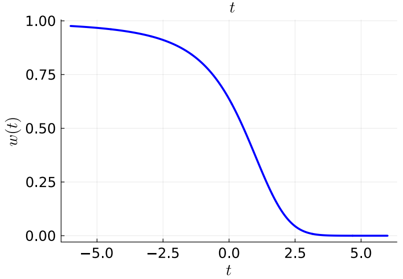
$$v(t) \coloneqq \frac{\mathcal{N}(t;0,1)}{F(t;0,1)}$$

Converges to $-t$ as $t \to -\infty$

$$w(t) \coloneqq v(t) \cdot [v(t) + t]$$

- This can be generalized to an arbitrary interval $[a, b]$ where the Gaussian is truncated!





Decision Making: Match Quality and Leaderboards

- **Match Quality**: Decide if two players i and j should be matched
  - **Idea**: Pick the pair (i, j) where the two players have equal skills

$$\text{Quality}(i,j) = \frac{P\left(p_i \approx p_j | \mu_i - \mu_j, \sigma_i^2 + \sigma_j^2\right)}{P\left(p_i \approx p_j | \mu_i - \mu_j = 0, \sigma_i^2 + \sigma_j^2 = 0\right)}$$

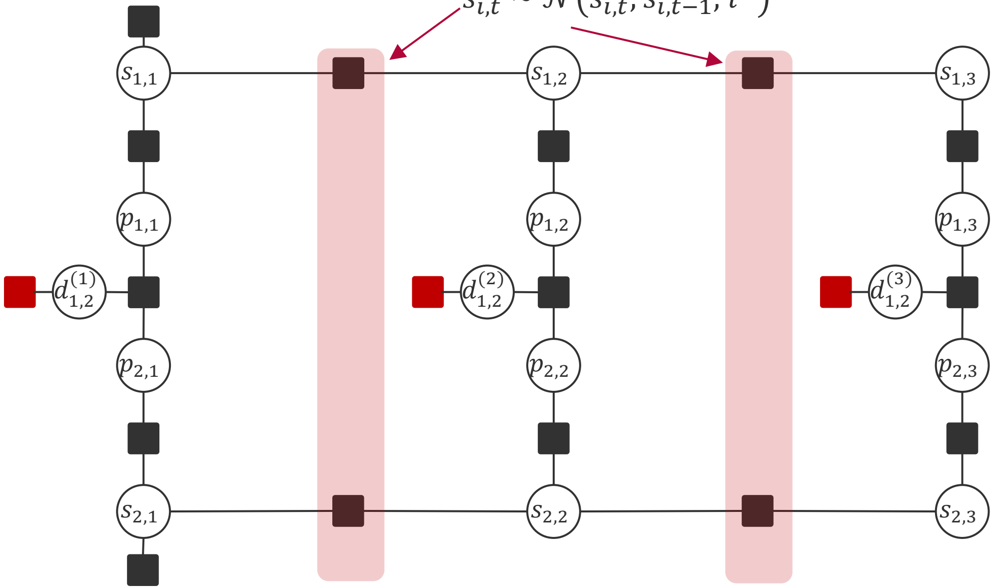
  - **Observation**: This pair (i, j) approximately maximizes the information (entropy!) of the predicted match outcome because it gets closest to 50% winning probability

- **Leaderboard**: Decide how to display the best to worst player
  - **Observation**: There is an asymmetry in making a ranking mistake
    - **Cheap**: Ranking a truly good player lower than they should be (why?)
    - **Expensive**: Ranking a truly bad player higher than they should be (why?)
    - The loss minimizer of this decision process is a **quantile** $\mu - k \cdot \sigma$

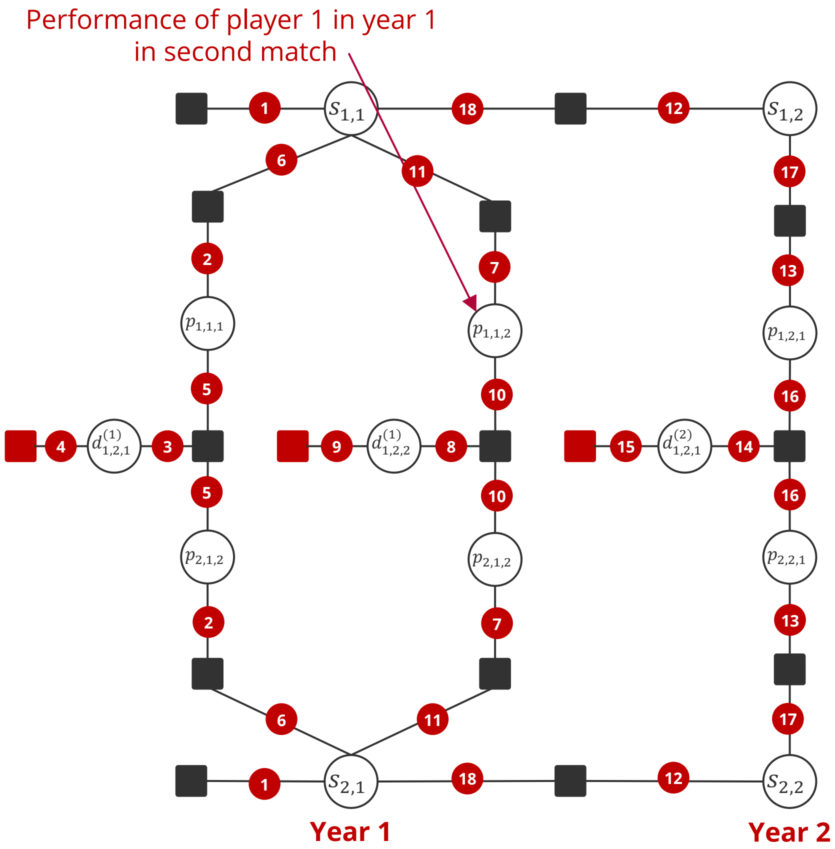
Skill Dynamics

- **Dynamics**: In reality, skills of players evolve over time and are not stationary
  - **Idea**: Since we do not know which direction, assume that the skill of player i at time t depends on the skill of the same player at time $t-1$ via

$$s_{i,t} \sim \mathcal{N}\left(s_{i,t}; s_{i,t-1}, \tau^2\right)$$

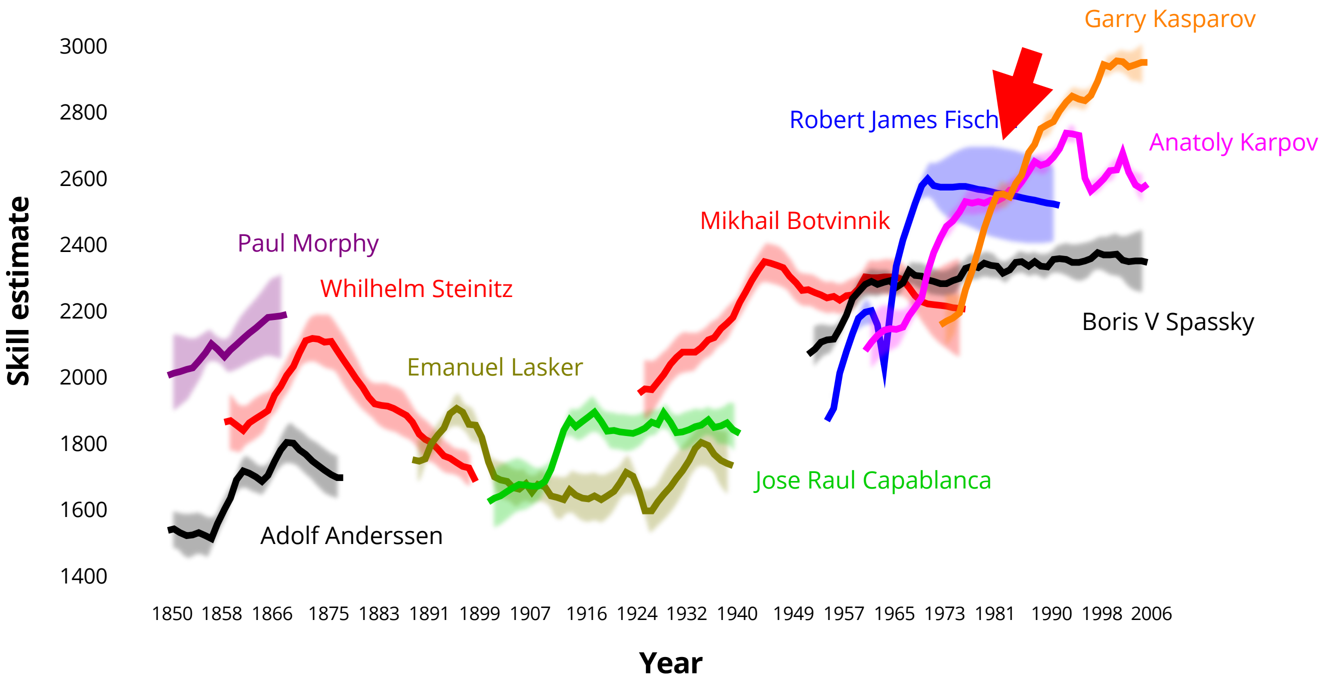


TrueSkill Through Time: Message Schedule



- Four Phases
  1. **Prior (1)**: Send prior messages to each skill variable for the first year of a player
  2. **Annual Matches (2-11)**: Loop over all (2-player) matches in a year until the skill marginals for all active player in that year does not change (much) anymore
  3. **Forward Dynamics (12)**: Send skill dynamics messages forward in time from t to $t + 1$ and keep running step 2. (13 – 17).
  4. **Backward Dynamics (18)**: Send skill dynamics messages backward in time from year $t + 1$ to t and keep running step 2. (2-11)
- Stop when no variable in the outer loop changes much anymore.

TrueSkill-Through-Time: Chess Players



History of Chess
3.5M match outcomes
20 million variables
40 million factors

Overview

Motivating Example: Information and Coin Tosses

- **Scenario 1**:
  - A coin toss with uncertain outcome modelled via $X \sim \mathrm{Ber}(p)$
  - $h(x;p)$ is the information/surprise received when you observe the value of x
  - **Question**:
    - How much is $h(1;1)$ when the success probability was 100%?
    - What's the relation between $h(1; p = 99\%)$ and $h(1; q = 1\%)$?
  - **Conclusion**: $h(x)$ is monotonically decreasing in $p(x)$
- **Scenario 2**:
  - Two independent coins are tossed modelled via $p(x,y) = p(x) \cdot p(y)$
  - **Question**: In what relation does $h(x,y)$ stand to $h(x)$ and $h(y)$?
  - **Conclusion**: If $p(x,y) = p(x) \cdot p(y)$ then $h(x,y) = h(x) + h(y)$

$$h(x) = -\log_b\big(p(x)\big)$$

$\boldsymbol{h(x,y) = h(x) + h(y)}$

$\boldsymbol{h(x,y) > h(x) + h(y)}$

$\boldsymbol{h(x,y) < h(x) + h(y)}$

Measure of Information: Entropy

- **Entropy**. *The* entropy *of a random variable X is the* average level of information *inherent to the variables outcomes and is defined by ($b > 1$)*

$$H_b[X] := -\sum_x P(X=x) \cdot \log_b\big(P(X=x)\big)$$
$$= E_{x\sim P}\big[-\log_b\big(p(x)\big)\big]$$

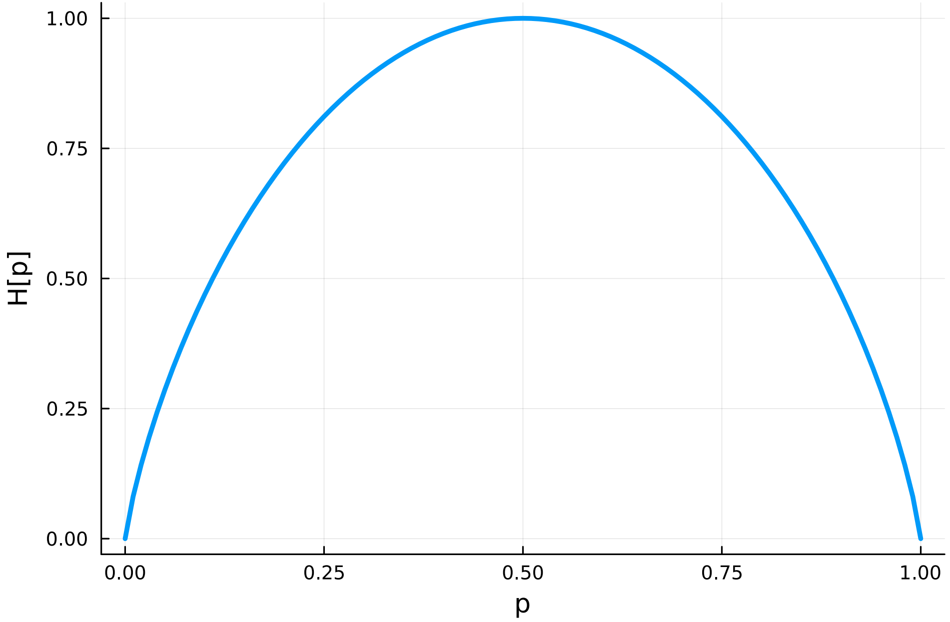
- **Khinchin (1957)**. *Entropy $H[X]$ as a* measure of information *of a random variable X follows from the following four axioms:*
  1. *$H[X]$ depends only on the probability distribution of X.*
  2. *$H[X]$ is maximal for the uniform distribution $P(X)$.*
  3. *$H[Y] = H[X]$ if X and Y have the same non-zero probabilities.*
  4. *For any random variables X and Y,*

$$H[X,Y] = H[X] + \underbrace{\sum_x P(X=x) \cdot\ H[Y \mid X=x]}_{H[Y|X]}$$

Aleksandr Khinchin (1894 – 1959)

Example: Binary Entropy

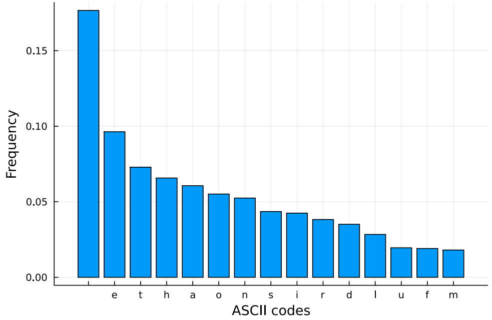
$$H_2[p] = p \cdot \log_2(p) + (1 - p) \cdot \log_2(1 - p)$$

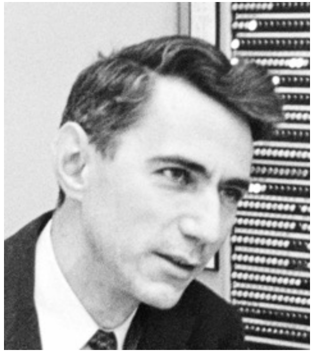


Entropy and the Noiseless Coding Theorem

- **(Shannon 1948).** *N independent and identically distributed random variables each with entropy $H[X]$ can be compressed into more than $N \cdot H[X]$ bits with negligible risk of information loss, as $N \to \infty$; but if they are compressed into fewer than $N \cdot H[X]$ bits it is virtually certain that information will be lost.*
- **Application** in data compression when modelling the value X of a byte modelled as a random variable over $n = 256$ values
  - **Random bytes**: $H[X] = -\sum_{i=1}^{256} \frac{1}{256} \log_2\left(\frac{1}{256}\right) = -\log_2\left(\frac{1}{256}\right) = 8$
  - **Random letters from the English alphabet**:

$$H = 4.48917$$





Claude Shannon
(1913 – 2001)

Noiseless Coding Theorem: An Example

- **Scenario**: We have 8 class labels with probabilities $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right\}$
- **Naïve Encoding**: We use a uniform distribution with 3 bits per symbol
$$H\left[\left\{\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right\}\right] = 3$$
- **However**, the entropy is 2 bits!
$$H[X] = 2$$
- **Prefix Code**: Unique binary prefix of consecutive 1's for each unique probability
- **Decode**: $\underbrace{1\ 1\ 0}_{C_3}\ \underbrace{0}_{C_1}\ \underbrace{1\ 1\ 1\ 0}_{C_4}$

Class	Code	$P(C)$	Length	$E[\text{Length}]$
1	0	1/2	1	16/32
2	10	1/4	2	16/32
3	110	1/8	3	12/32
4	1110	1/16	4	8/32
5	111100	1/64	6	3/32
6	111101	1/64	6	3/32
7	111110	1/64	6	3/32
8	111111	1/64	6	3/32

Viel Spaß bis zur nächsten Vorlesung!