The Sum-Product Algorithm Explained

Ralf Herbrich
Amazon

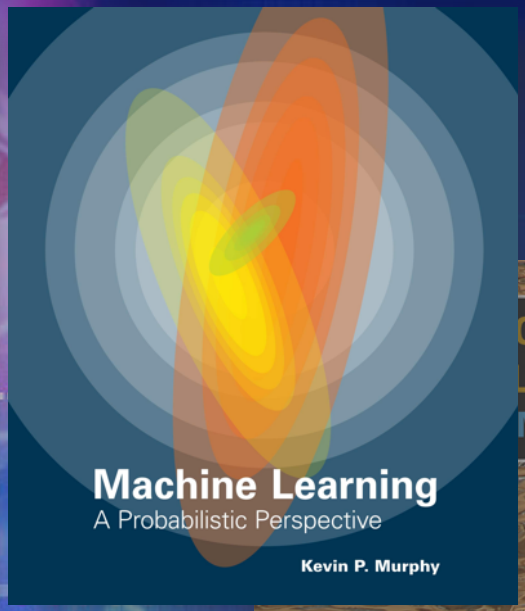
Overview

- Theory
  - **Inference in Factor Graphs**
  - **Approximate Message Passing**
- Applications @ Microsoft
  - **TrueSkill: Gamer Rating and Matchmaking**
  - **TrueSkill Through Time: History of Chess**

Background Material

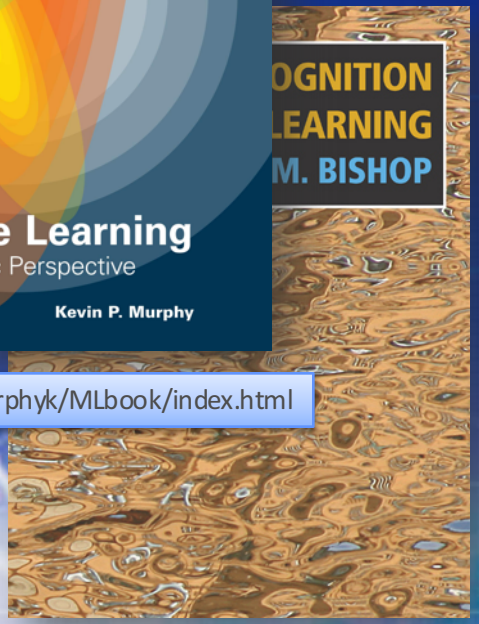
Coursera

http://www.coursera.org



http://www.cs.ubc.ca/~murphyk/MLbook/index.html

http://www.cs.ucl.ac.uk/staff/d.barber/brml/



http://research.microsoft.com/en-us/um/people/cmbishop/PRML/index.htm

Overview

- **Theory**
  - **Inference in Factor Graphs**
  - **Approximate Message Passing**
- Applications
  - **TrueSkill: Gamer Rating and Matchmaking**
  - **TrueSkill Through Time: History of Chess**

Graphical Models

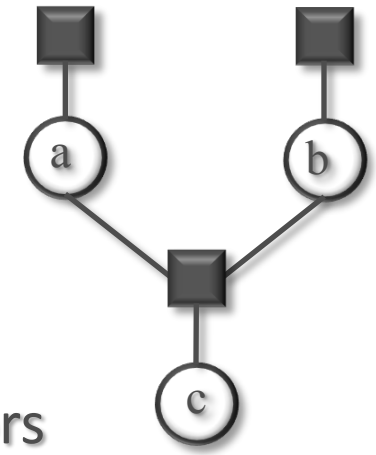
- **Definition:** Graphical representation of joint probability distribution
  - Nodes: ◯ = Variables
  - Edges: Relationship between variables
- **Variables**:
  - Observed Variables: Data
  - Unobserved Variables: 'Causes' + Temporary/Latent
- **Key Questions**:
  - (Conditional) *Dependency*: $p(a,b|c) \stackrel{?}{=} p(a|c) \cdot p(b|c)$
  - *Inference*/Marginalisation: $p(a,b) = \sum_c p(a,b,c)$

Factor Graphs

- **Definition:** Graphical representation of product structure of a function (Wiberg, 1996)
  - Nodes: ■ = Factors ◯ = Variables
  - Edges: Dependencies of factors on variables.

- **Semantic:**

$$p(\mathbf{x}) = \prod_f f\left(\mathbf{x}_{V(f)}\right)$$



  - Local variable dependency of factors

$$p(a,b,c) = f_1(a) \cdot f_2(b) \cdot f_3(a,b,c)$$

Factor Graphs and Bayes' Law

- Bayes' law

$$p(\mathbf{s}|y) \propto p(y|\mathbf{s}) \cdot p(\mathbf{s})$$

- Factorising prior

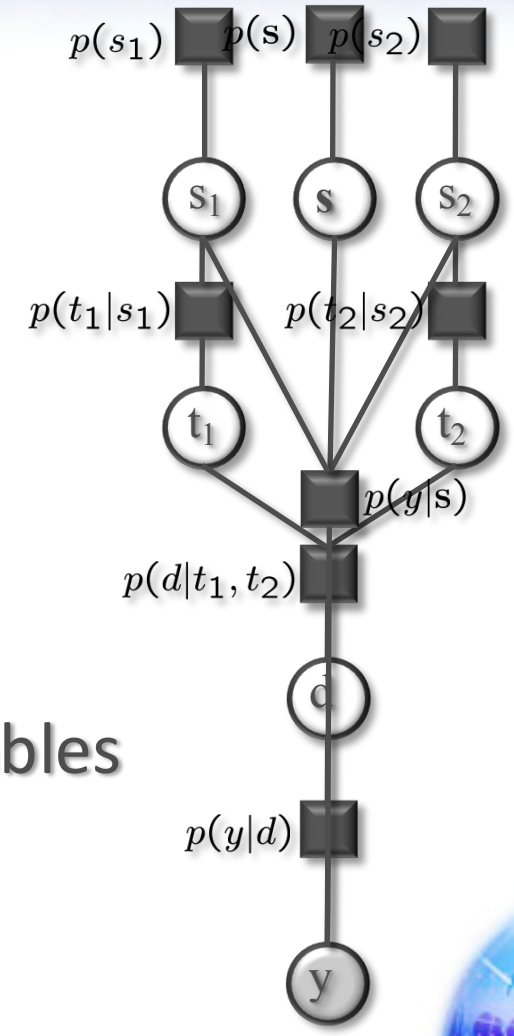
$$p(\mathbf{s}) = p(s_1) \cdot p(s_2)$$

- Factorising likelihood

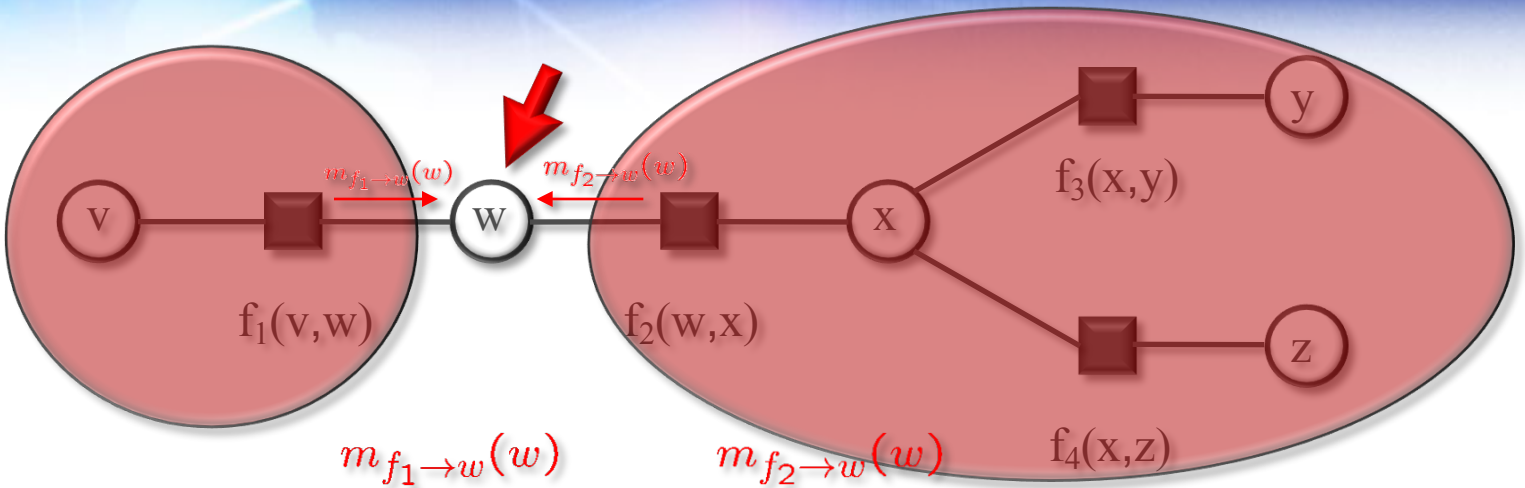
$$p(y, \mathbf{t}, d|\mathbf{s}) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d)$$

- Inference: Sum out latent variables

$$p(y|\mathbf{s}) = \sum_{\mathbf{t}} \sum_{d} p(y, \mathbf{t}, d|\mathbf{s})$$



Factor Trees: Separation

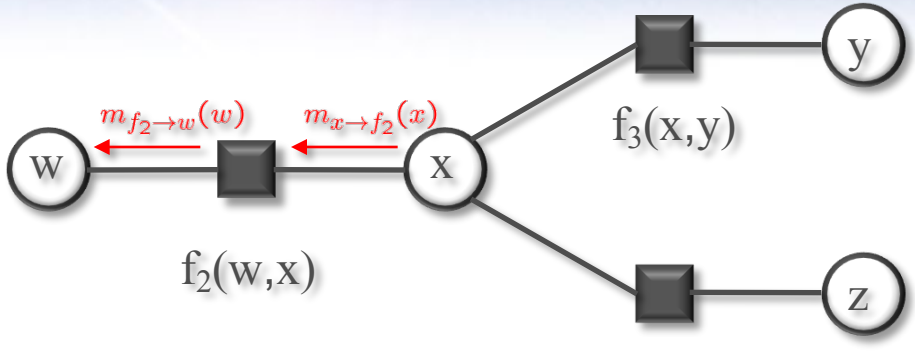


$$p(w) = \left[\sum_{v} \sum_{x} f_1(v,w)\right] \left[\sum_{y} \sum_{z} f_2(w,x) f_3(x,y) f_4(x,z)\right]$$

$m_{f_1 \to w}(w)$ $m_{f_2 \to w}(w)$

Observation: Sum of products becomes product of sums of all messages from neighbouring factors to variable!

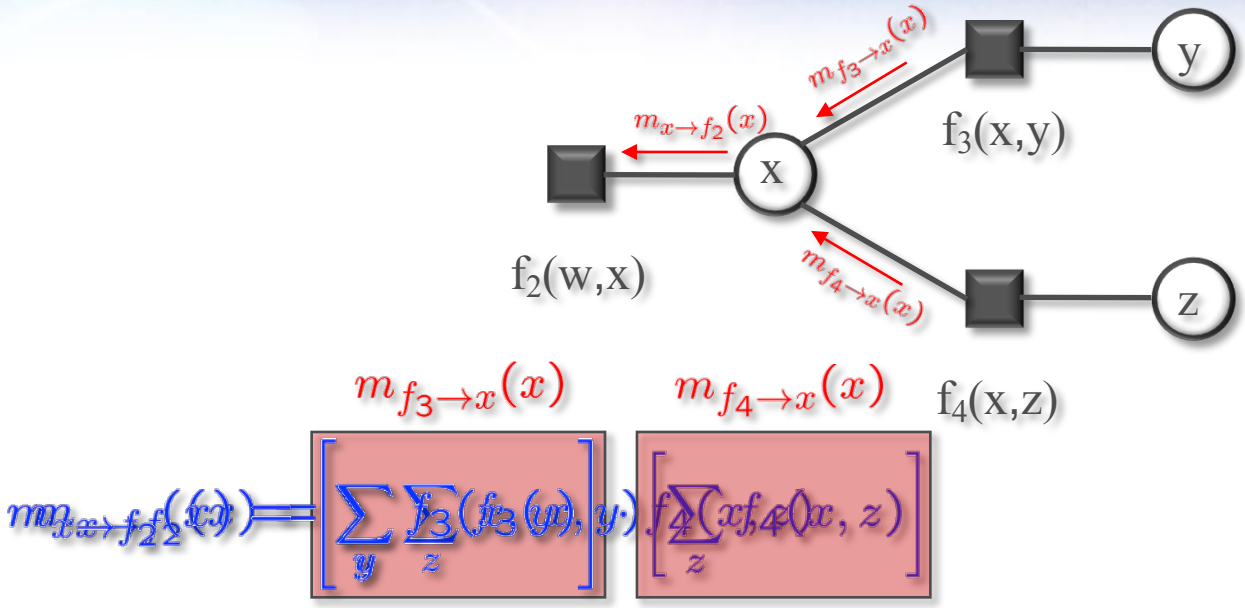
Messages: From Factors To Variables



$$m_{f_2 \to w}(w) = \sum_x f_2(w,x) \underbrace{\left[\sum_y f_3(x,y) \sum_z f_4(x,z) \right]}_{m_{x \to f_2}(x)}$$

Observation: Factors only need to sum out all their local variables!

Messages: From Variables To Factors



Observation: Variables pass on the product of all incoming messages!

The Sum-Product Algorithm

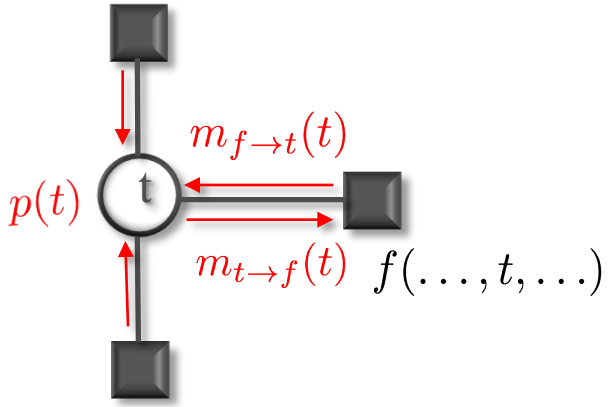
- Three update equations (Aji & McEliece, 1997)

$$
\begin{aligned}
p(t) &= \prod_{f \in F_t} m_{f \to t}(t) \\
m_{f \to t_1}(t_1) &= \sum_{t_2} \sum_{t_3} \cdots \sum_{t_n} f(t_1, t_2, t_3, \ldots) \prod_{i>1} m_{t_i \to f}(t_i) \\
m_{t \to f}(t) &= \prod_{f_j \in F_t \setminus \{f\}} m_{f_j \to t}(t)
\end{aligned}
$$

- Update equations can be directly derived from the distributive law.
- Calculate all marginals at the same time!
- Only need to pass messages twice along each edge!

Practical Considerations II

- **Redundant computations:**



$$p(t) = \prod_{f\in F_t} m_{f\to t}(t)$$

$$m_{t\to f}(t) = \prod_{f_j\in F_t\setminus\{f\}} m_{f_j\to t}(t) \quad \Rightarrow \quad p(t) = m_{t\to f}(t)\cdot m_{f\to t}(t)$$

- **Caching**: Only store $p(t)$ and $m_{f\to t}(t)$, then

$$m_{t\to f}(t) = \frac{p(t)}{m_{f\to t}(t)}$$

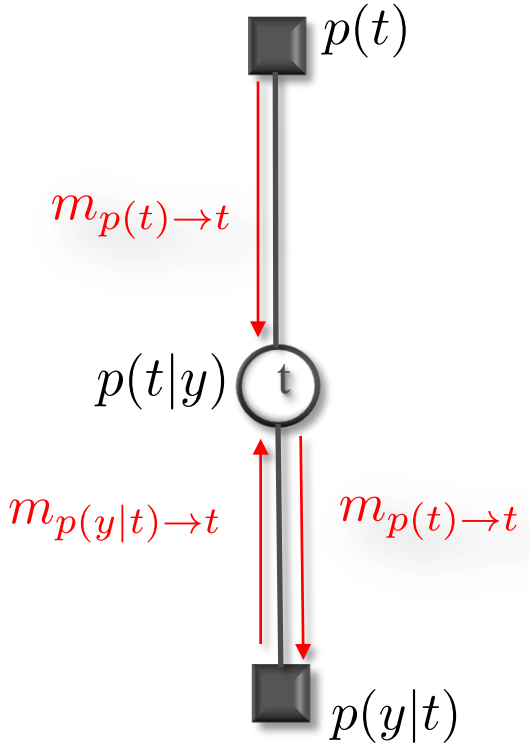
A Bayesian Interpretation

- **Recall Bayes' Law:**

$$p(t|y) \propto p(y|t) \cdot p(t)$$

- **Prior and Data Messages:**

$$p(t|y) \propto m_{p(y|t)\to t} \cdot m_{p(t)\to t}$$
$$\propto m_{p(y|t)\to t} \cdot m_{t\to p(y|t)}$$



Message passing is separating the likelihood and prior into outgoing and incoming message!

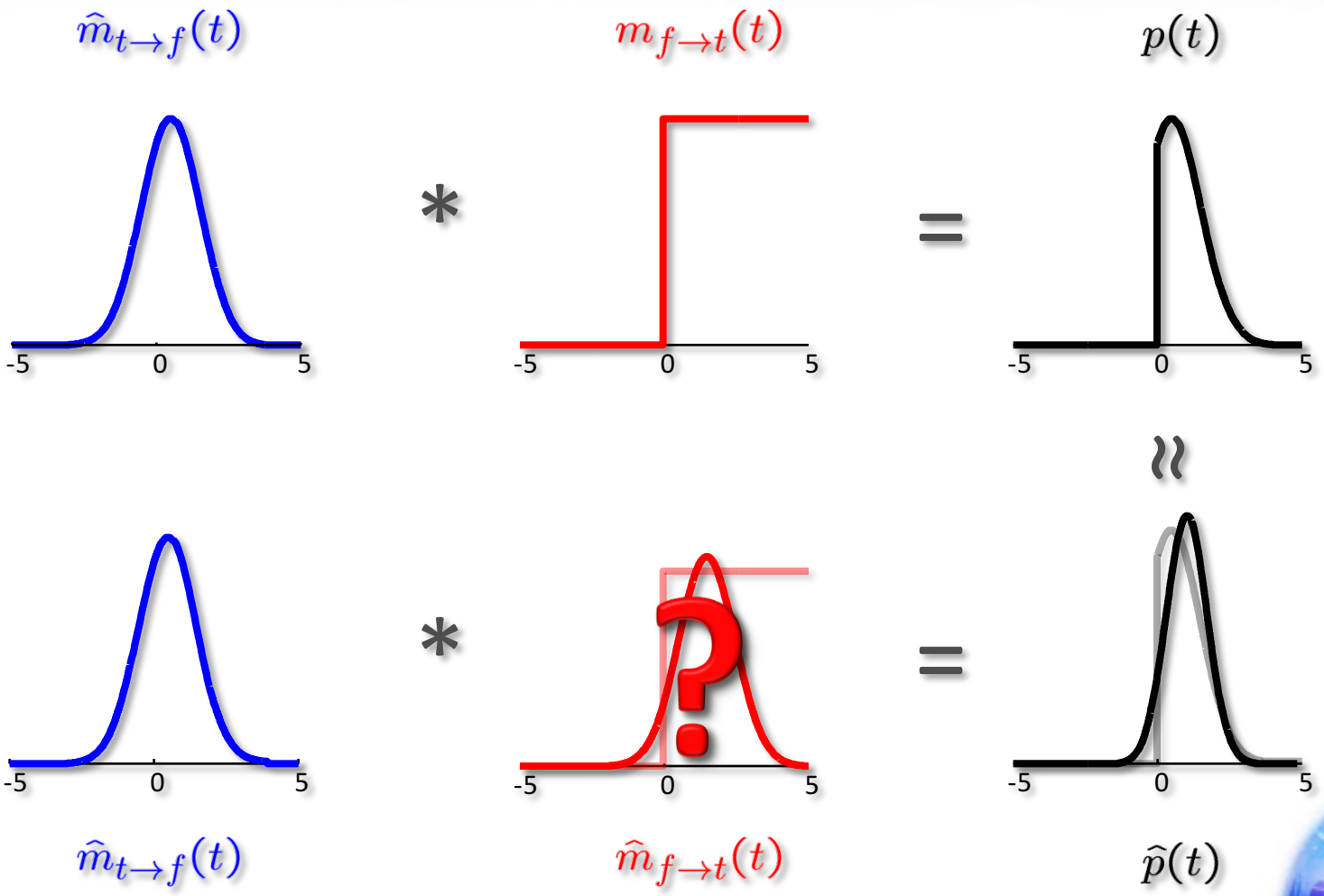
Approximate Message Passing

- **Problem:** The exact messages from factors to variables may not be closed under products.
- **Solution:** Approximate *each* marginal as well as possible in using a divergence measure on beliefs.
- **General Idea:** Leave-one out approximation

$$\widehat{p}(t) = \operatorname{argmin}_{\widehat{p}} , D\left[\overbrace{m_{f\to t} \cdot \widehat{m}_{t\to f}}^{p(t)}, \widehat{p} \right]$$

$$\widehat{m}_{f\to t}(t) = \frac{\widehat{p}(t)}{\widehat{m}_{t\to f}(t)}$$

Approximate Message Passing



Divergence Measures

- **Kullback-Leibler Divergence:** Expected log-odd ratio between two distributions:

$$\mathrm{KL}(p,q) := \sum_t p(t) \log \left(\frac{p(t)}{q(t)} \right)$$

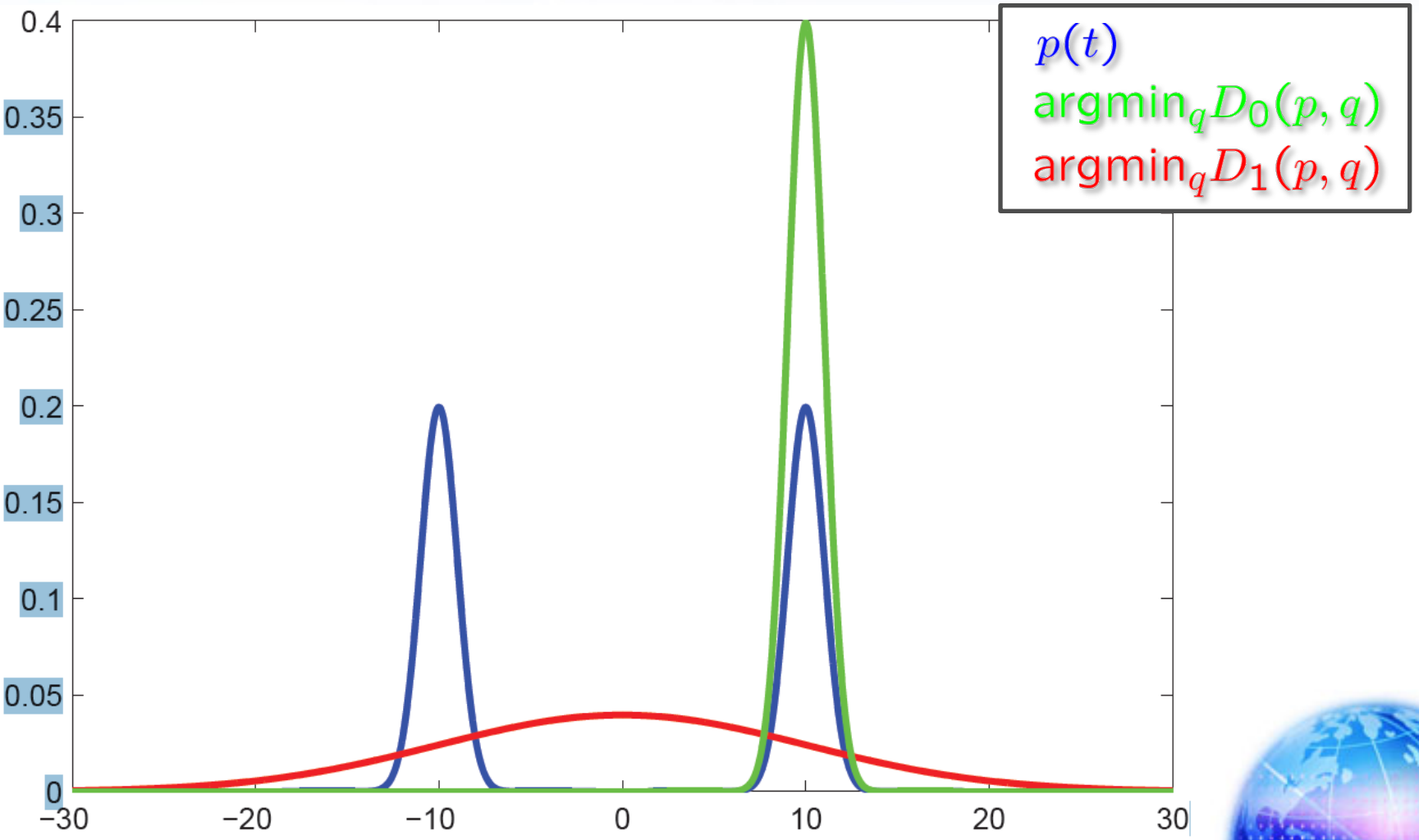
- **Minimizer for Exponential Families:** Matching the moments of the distribution $p(t)$!
- **General α-Divergence:**

$$D_\alpha(p,q) := \frac{1 - \sum_t \frac{p^{\alpha-1}(t)}{q^{\alpha-1}(t)}}{\alpha(1-\alpha)}$$

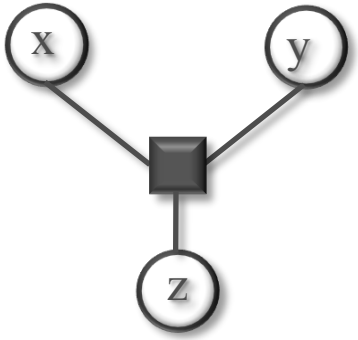
- **Special Cases:**

$$\begin{aligned} D_0(p,q) &= \mathrm{KL}(q,p) \\ D_1(p,q) &= \mathrm{KL}(p,q) \end{aligned}$$

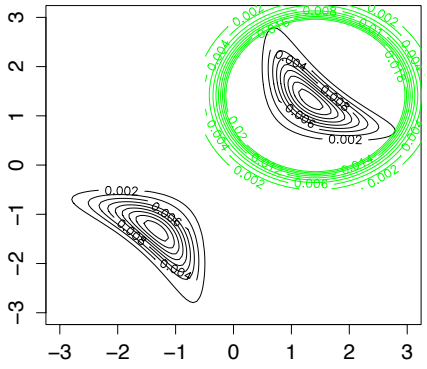
α-Divergence in Pictures

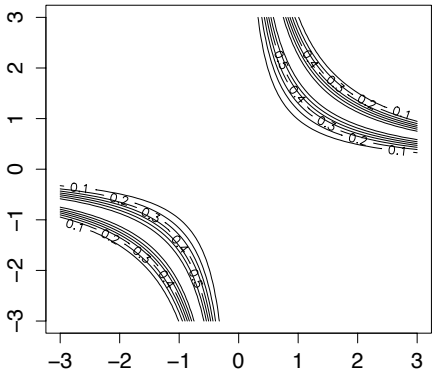


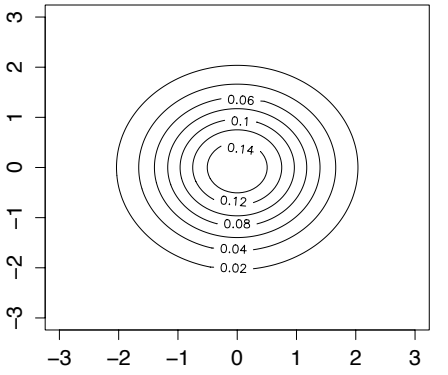
When to use which α-Divergence?



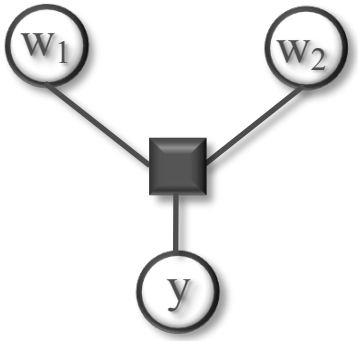
α=0 resolves multi-modality in the posterior at the expense of too much certainty!



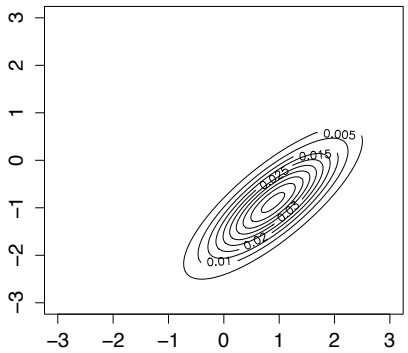


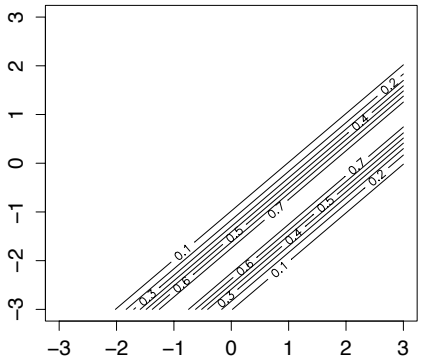


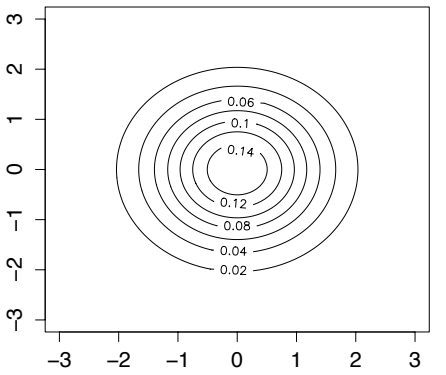
When to use which α-Divergence?



α=1 captures all uncertainty for uni-modal posterior distributions!

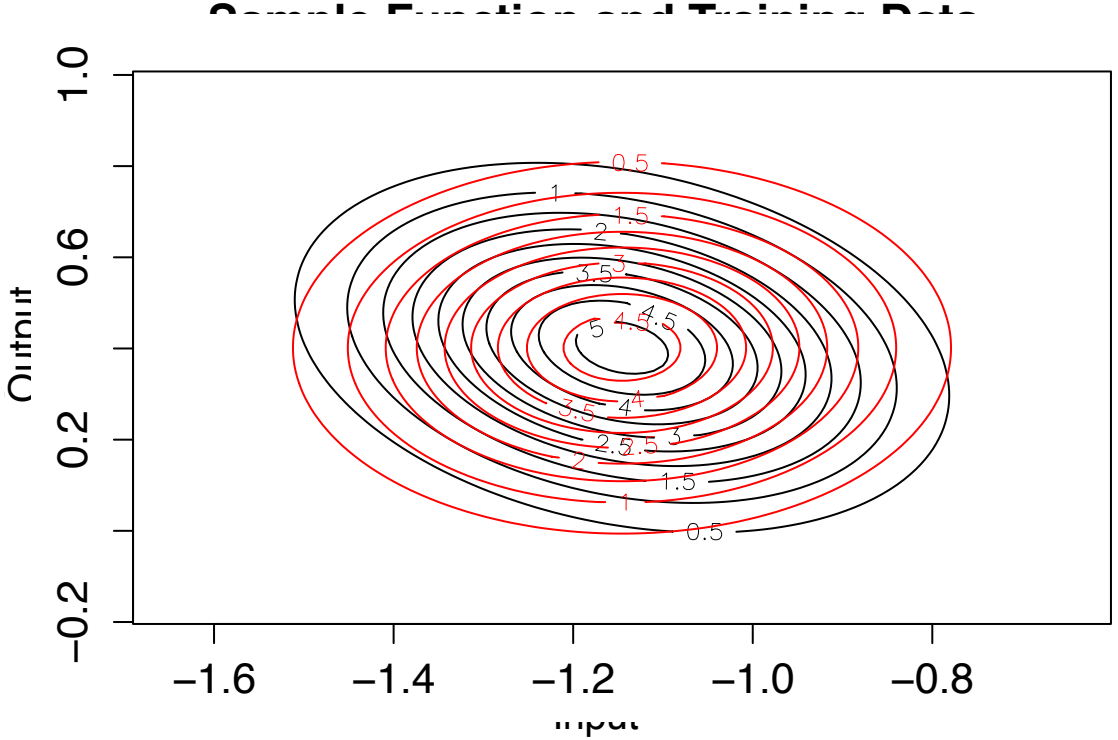






Sample (ctd)

$$p(\mathbf{w}|y,\mathbf{x}) = \mathcal{N}(y;[\sin(3t);\sin(t)\cos(6t)]^{\mathrm{T}}\mathbf{w},0.1)\cdot\mathcal{N}(\mathbf{w};\mathbf{0},\mathbf{I})$$



Overview

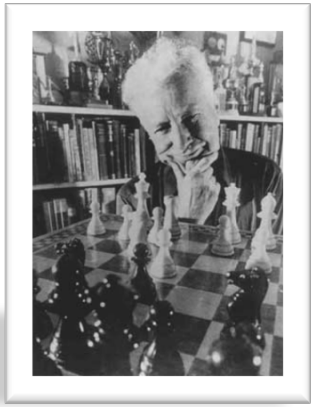
- Theory
  - Inference in Factor Graphs
  - Approximate Message Passing
- **Applications @ Microsoft**
  - TrueSkill: Gamer Rating and Matchmaking
  - TrueSkill Through Time: History of Chess

TrueSkill™

Joint work with Thore Graepel, Tom Minka & Phillip Trelford

Motivation

- Competition is central to our lives
  - Innate biological trait
  - Driving principle of many sports
- Chess Rating for fair competition
  - ELO: Developed in 1960 by Árpád Imre Élő
  - Matchmaking system for tournaments
- Challenges of online gaming
  - Learn from few match outcomes efficiently
  - Support multiple teams and multiple players per team

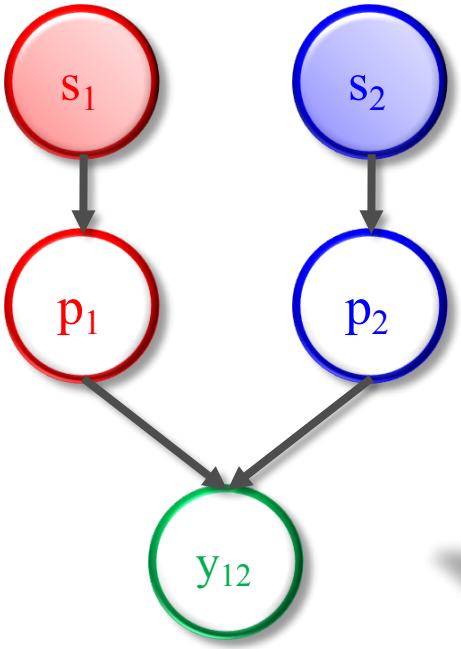


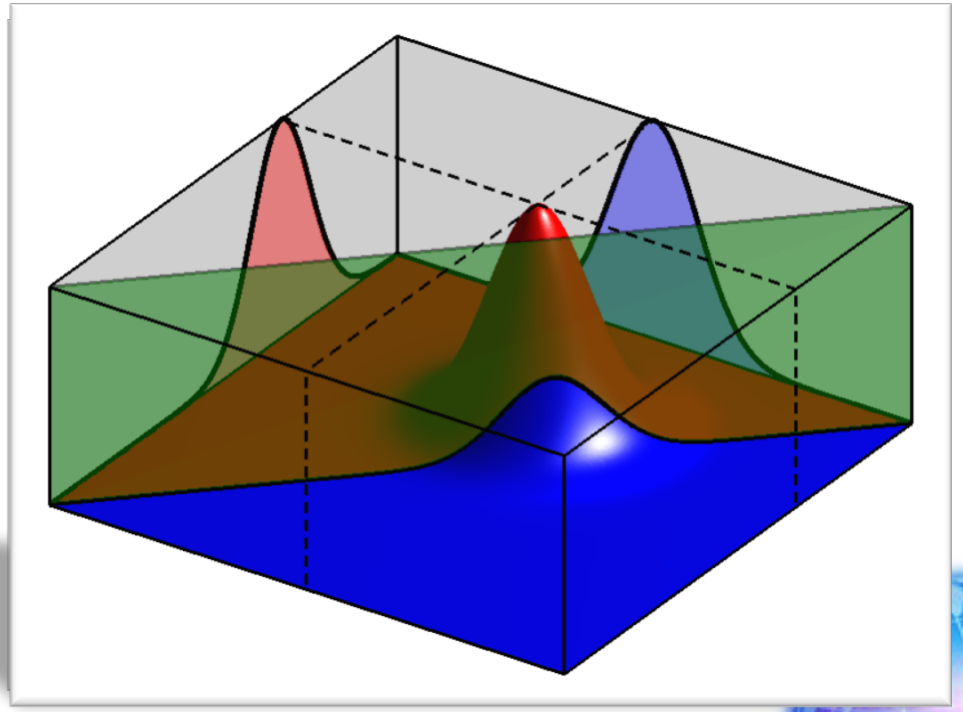
The Skill Rating Problem

- **Given:**
  - Match outcomes: Orderings among k teams
- **Qu**
  -

Two Player Match Outcome Model

- Latent Gaussian performance model for fixed skills
- Possible outcomes: Player 1 wins over 2 (and vice versa)

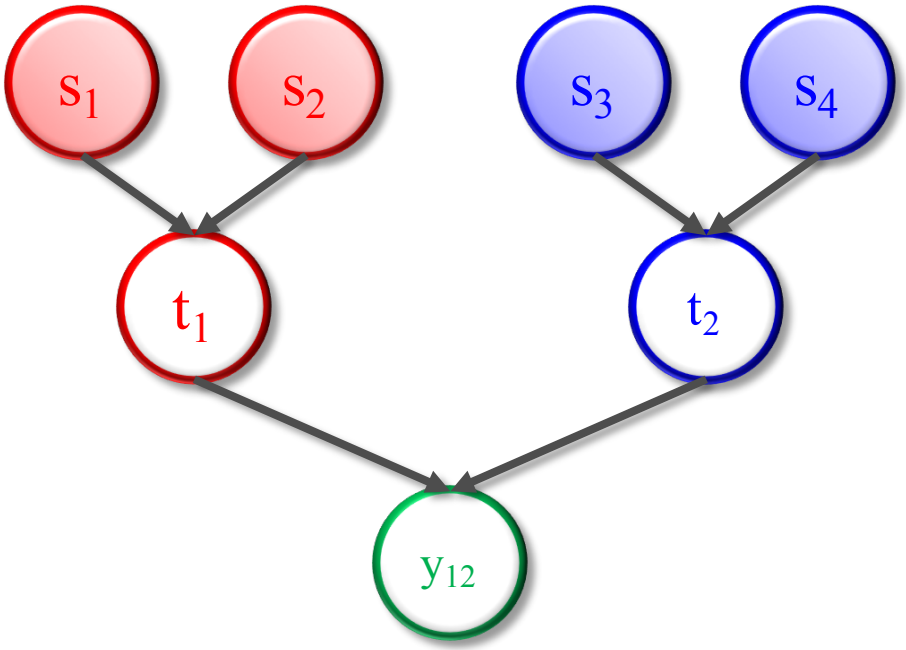




$$\mathbf{P}(y_{12} = (1,2)|p_1, p_2) = \mathbb{I}(p_1 > p_2)$$

Two Team Match Outcome Model

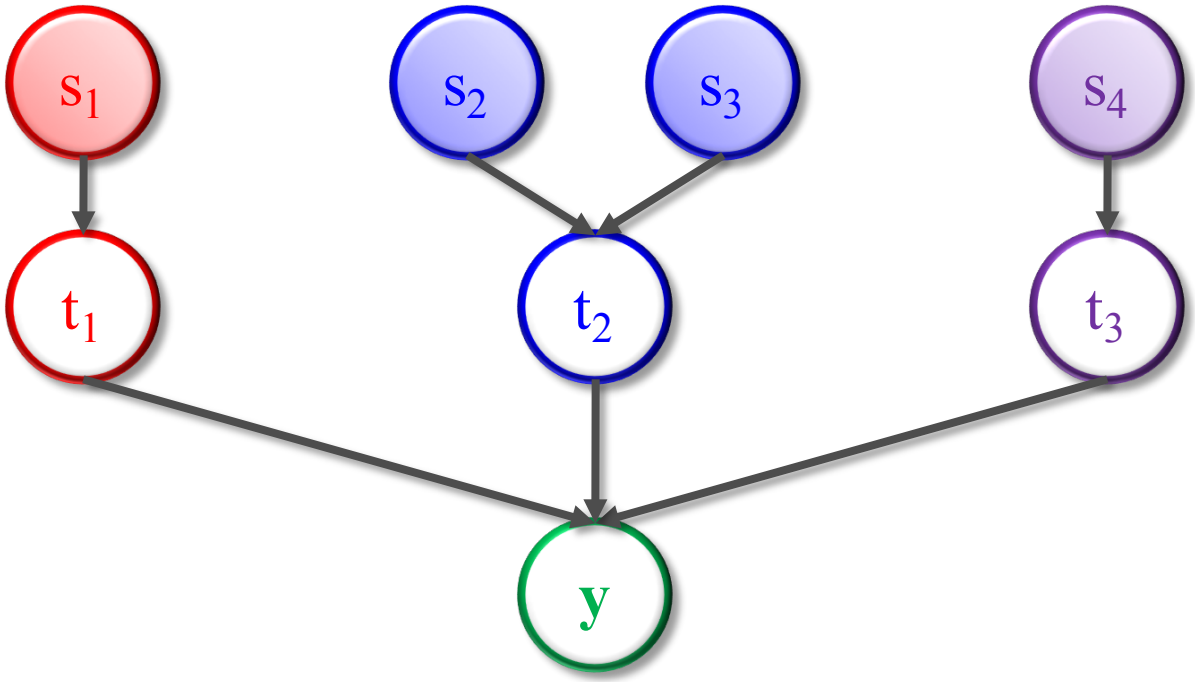
- Skill of a team is the sum of the skills of its members



$$\mathbf{P}(t_1|s_1, s_2) = \mathcal{N}\left(t_1; s_1 + s_2, 2 \cdot \beta^2\right)$$

Multiple Team Match Outcome Model

- Possible outcomes: Permutations of the teams

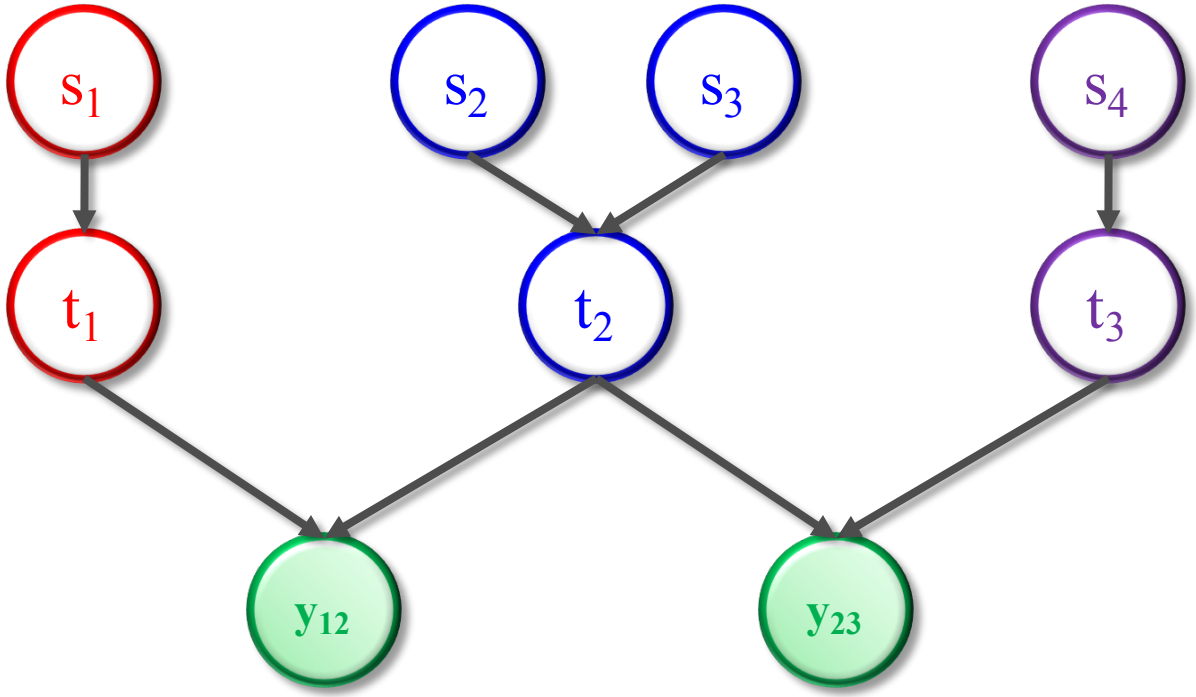


$$\mathbf{P}(\boldsymbol{y}|t_1, t_2, t_3) = \mathbb{I}(\boldsymbol{y} = (i, j, k)) \text{ where } t_i > t_j > t_k$$

Multiple Team Match Outcome Model

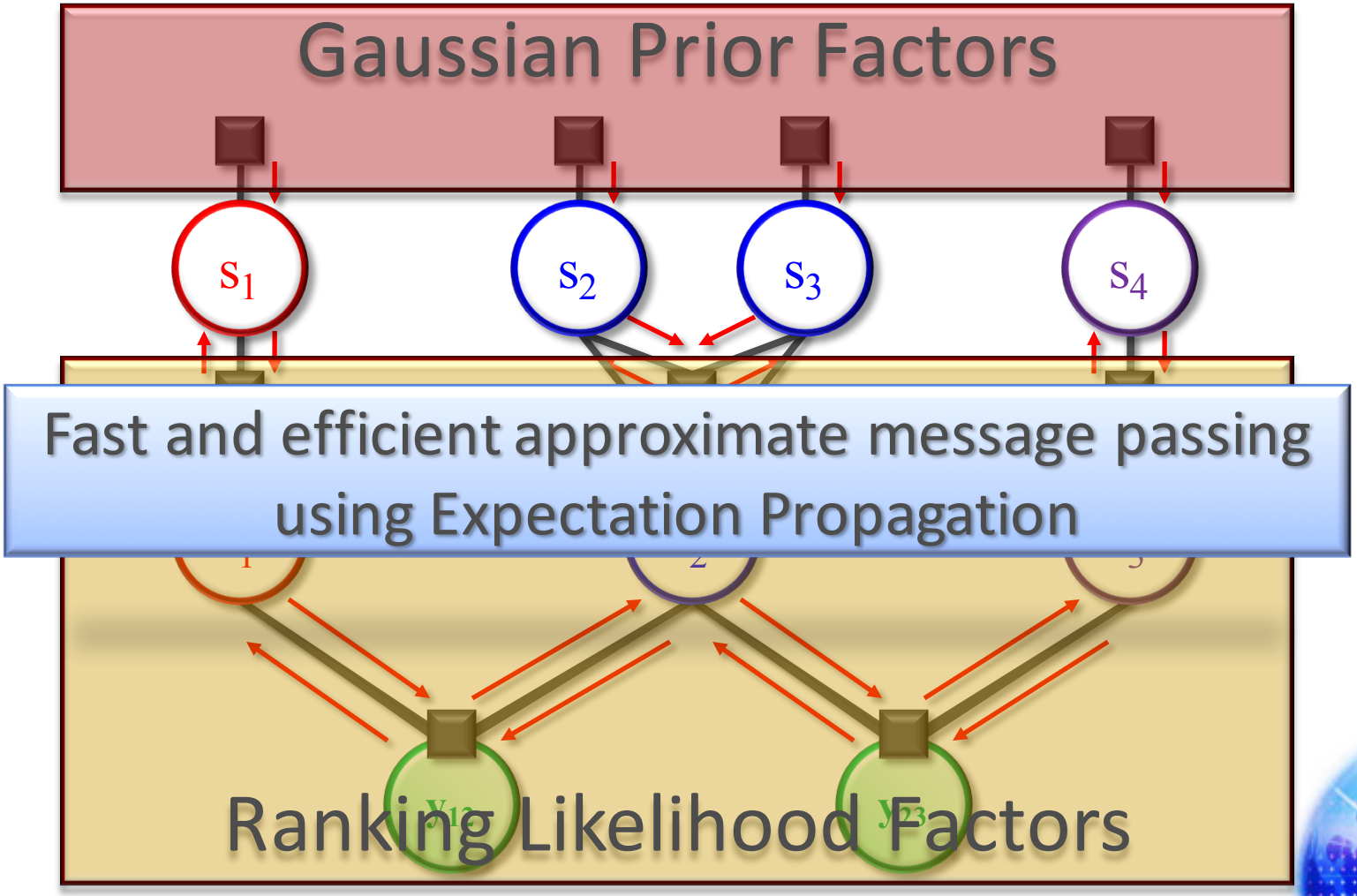
- But we are interested in the (Gaussian) posterior!

$$\mathbf{P}(s_i|\boldsymbol{y} = (1, 2, 3)) = \mathcal{N}(s_i; \mu_i, \sigma_i^2)$$



$y_{12} = (1, 2)$ $y_{23} = (2, 3)$

Efficient Approximate Inference

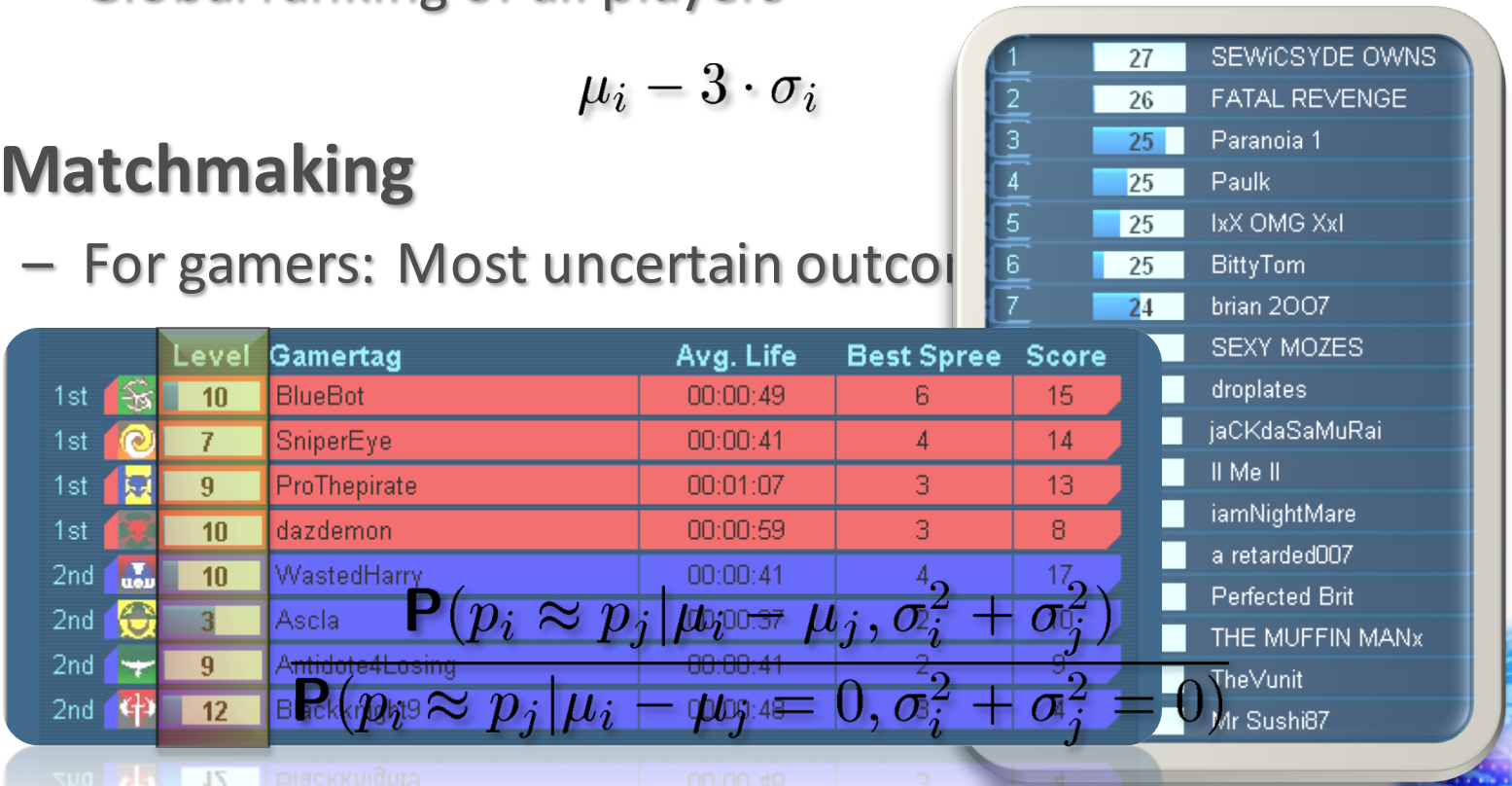


Applications to Online Gaming

- **Leaderboard**
  - Global ranking of all players

$$\mu_i - 3 \cdot \sigma_i$$

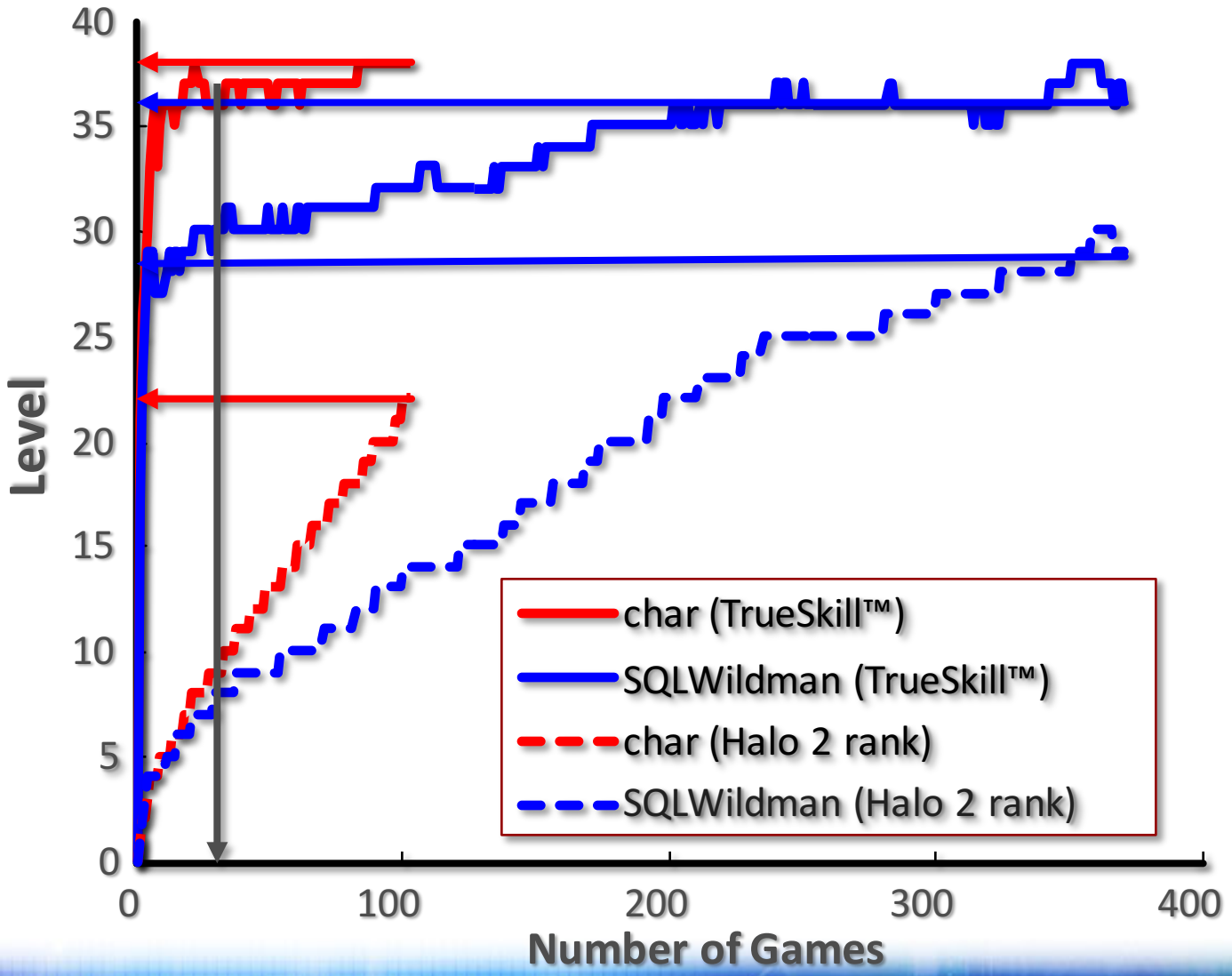
- **Matchmaking**
  - For gamers: Most uncertain outcome



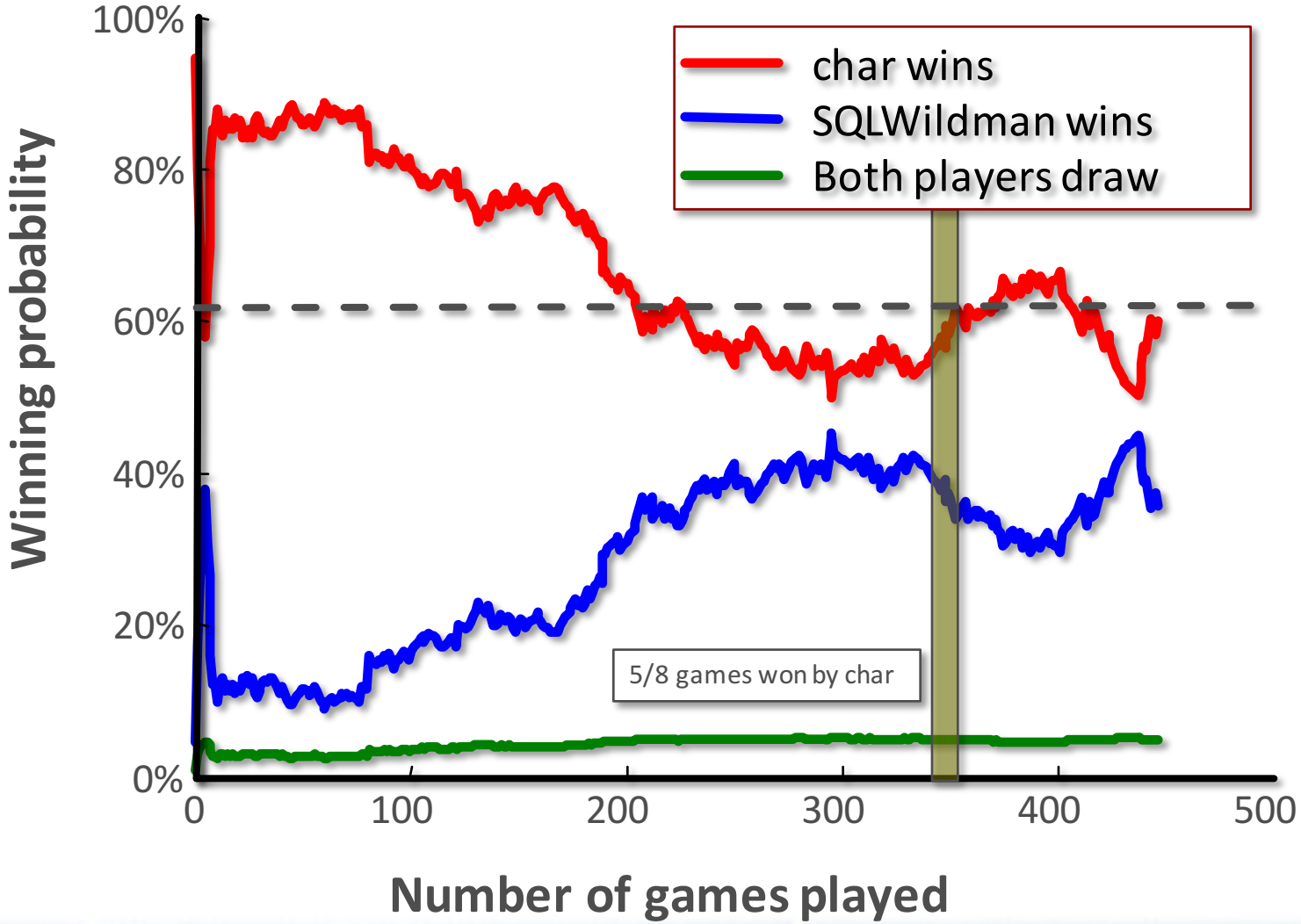
Experimental Setup

- **Data Set: Halo 2 Beta**
  - 3 game modes
    - Free-for-All
    - Two Teams
    - 1 vs. 1
  - > 60,000 match outcomes
  - ≈ 6,000 players
  - 6 weeks of game play
  - Publically available

Convergence Speed



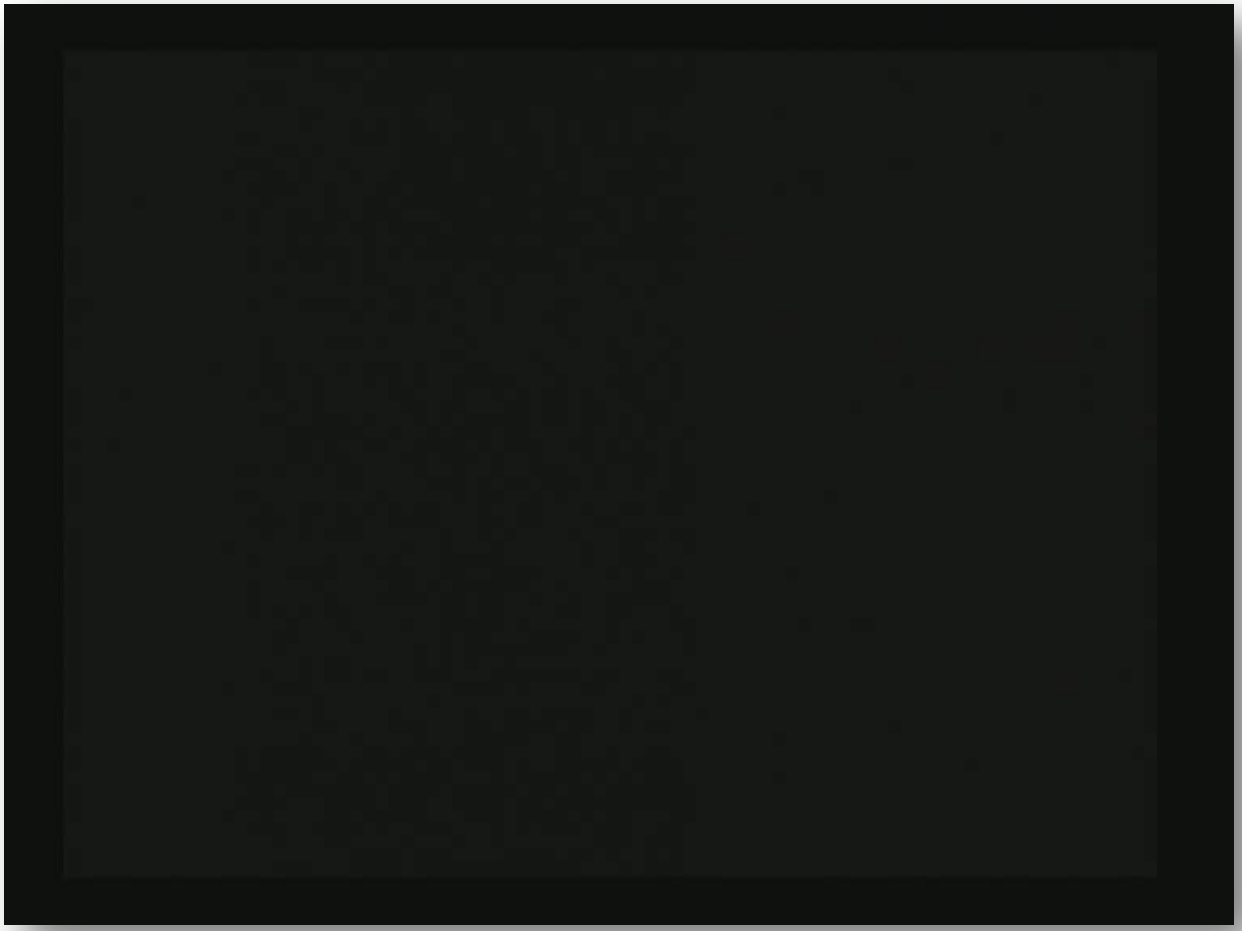
Convergence Speed (ctd.)



Xbox 360 & Halo 3

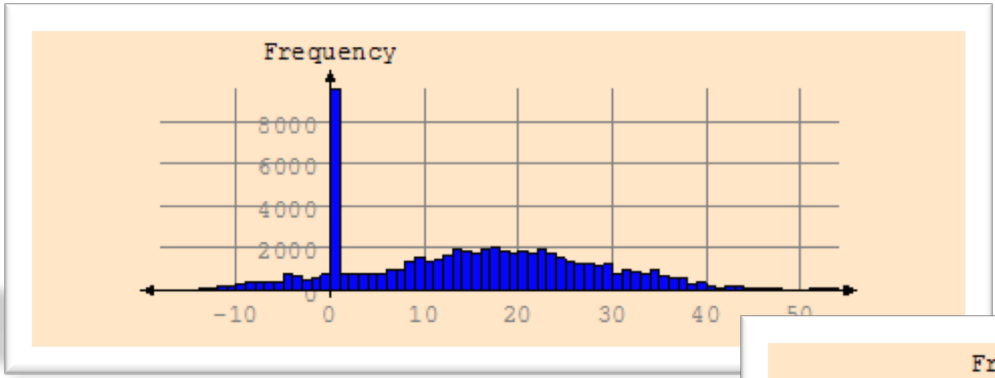
- **Xbox 360 Live**
  - Launched in September 2005
  - Every game uses TrueSkill™ to match players
  - > 10 million players
  - > 2 million matches per day
  - > 2 billion hours of gameplay
- **Halo 3**
  - Launched on 25th September 2007
  - Largest entertainment launch in history
  - > 200,000 player concurrently (peak: 1,000,000)

Halo 3 in Action

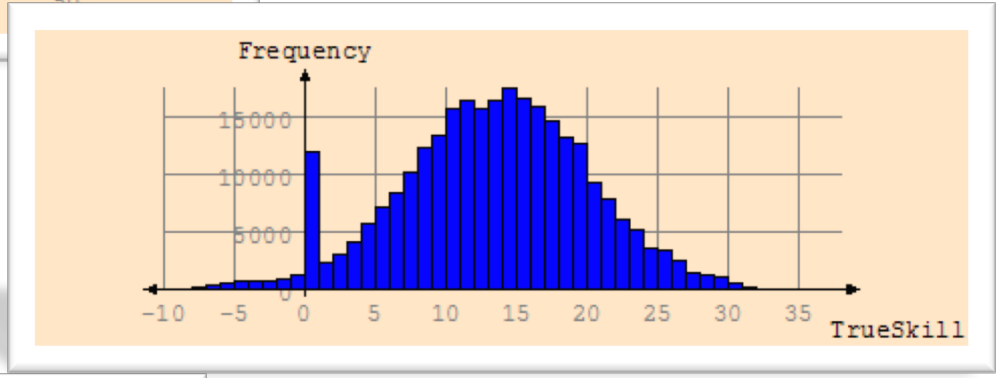


Halo 3 Public Beta Analysis

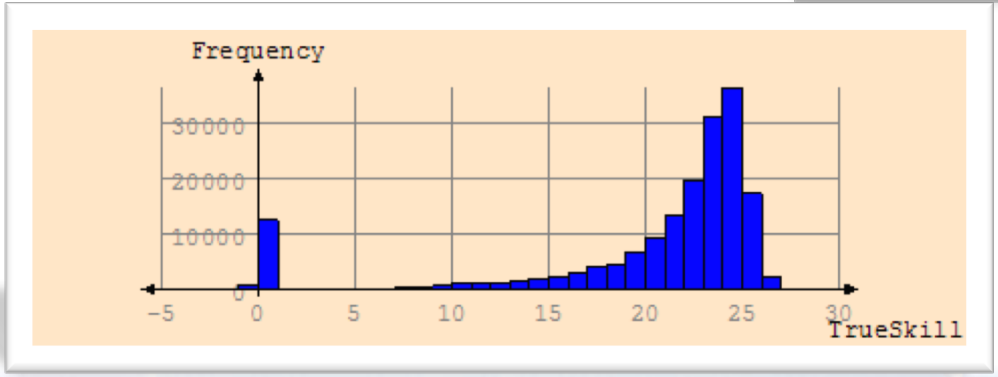
Skill Distributions of Online Games



Golf (18 holes): 60 levels



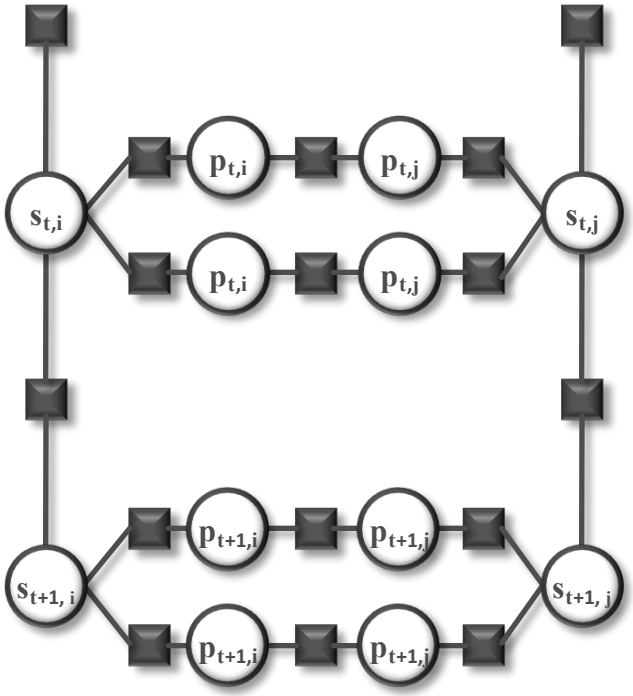
Car racing (3-4 laps): 40 levels



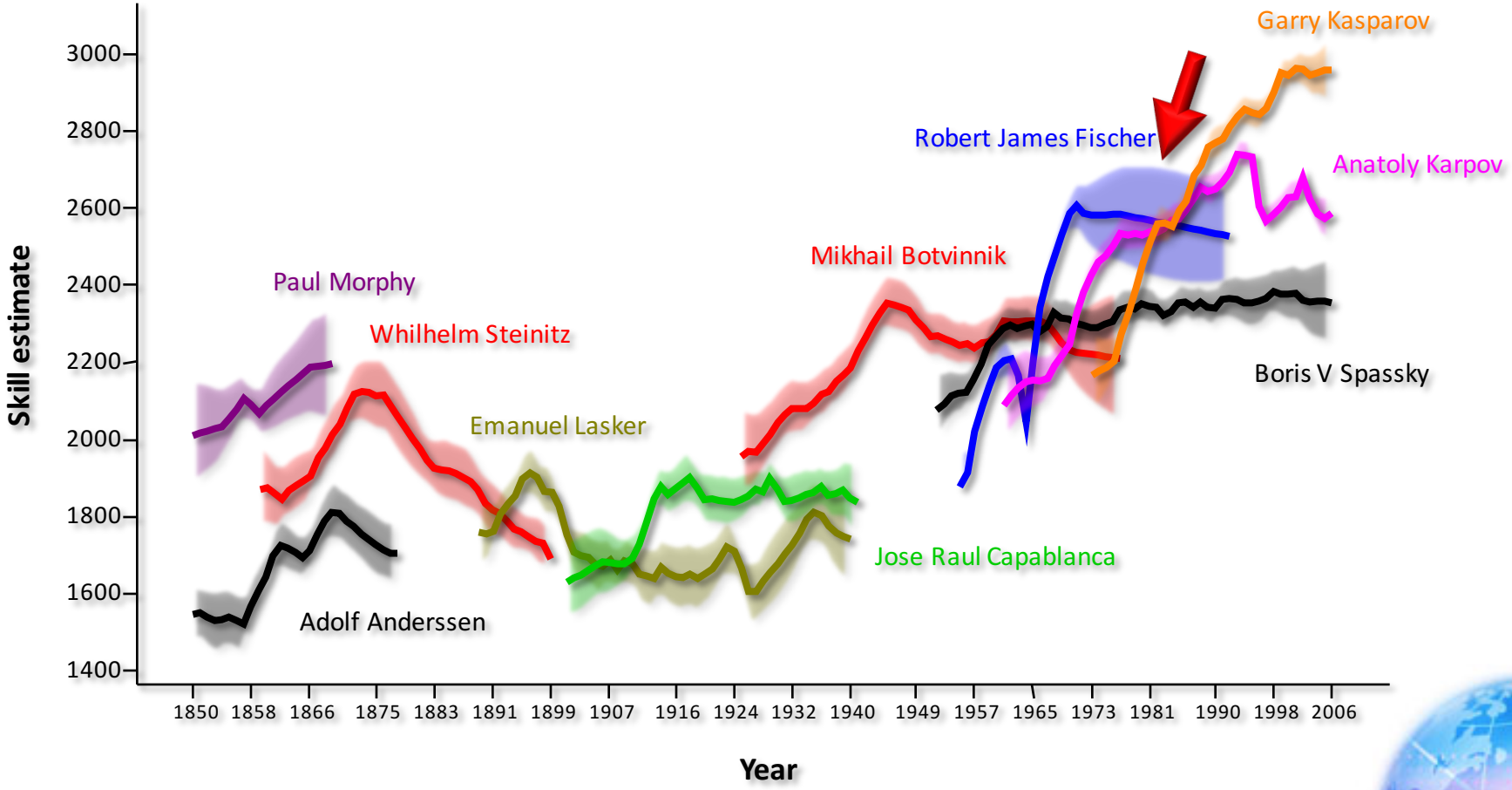
UNO (chance game): 10 levels

TrueSkill™ Through Time: Chess

- Model time-series of skills by smoothing across time
- History of Chess
  - 3.5M game outcomes (ChessBase)
  - 20 million variables (each of 200,000 players in each year of lifetime + latent variables)
  - 40 million factors



ChessBase Analysis: 1850 - 2006




Thanks!