Although all of the information in these slides has been published and is non-confidential, it may contain intellectual property owned by my prior employers.
Overview

• Background

• Technology Transfer Case Studies
  – TrueSkill: Gamer Rating and Matchmaking
  – Click-Through Rate Prediction in Online Advertising

• Technology Transfer Lessons
  – Process
  – Technical
Overview

• **Background**

• **Technology Transfer Case Studies**
  – TrueSkill: Gamer Rating and Matchmaking
  – Click-Through Rate Prediction in Online Advertising

• **Technology Transfer Lessons**
  – Process
  – Technical
1992 – 1997 (Berlin, Diploma)
1997 – 2000 (Berlin, PhD)
2000 – 2009 (Microsoft Research)
2009 – 2011 (Microsoft)
2011 – 2012 (Facebook)
2012 – Present (Amazon)
Overview

- Background

- **Technology Transfer Case Studies**
  - TrueSkill: Gamer Rating and Matchmaking
  - Click-Through Rate Prediction in Online Advertising

- Technology Transfer Lessons
  - Process
  - Technical
• **Definition (Wiki):** Process of moving promising research topics into a level of maturity ready for bulk manufacturing or production

• **Practice:** Often failing due to
  – Different Success Criteria (Product vs. Publication)
  – No Training Programs for Technology Transfer
  – Processes Are Hard to Generalize (Structure?)


**Given:**

- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

**Questions:**

- Skill $s_i$ for each player such that
  
  $$s_i > s_j \Rightarrow P(\text{Player } i \text{ wins}) > P(\text{Player } j \text{ wins})$$

- Global ranking among all players
- Fair matches between teams of players
• **Given:**
  - Match outcomes: Orderings among \( k \) teams consisting of \( n_1, n_2, \ldots, n_k \) players, respectively

• **Questions:**
  - Skill ratings \( s_i \) for each player such that
  - Global ranking among all players
  - Fair matches between teams of players

### The Skill Rating Problem

<table>
<thead>
<tr>
<th>Level</th>
<th>Gamertag</th>
<th>Avg. Life</th>
<th>Best Spree</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>BlueBot</td>
<td>00:00:49</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>1st</td>
<td>SniperEye</td>
<td>00:00:41</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>1st</td>
<td>ProThepirate</td>
<td>00:01:07</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>1st</td>
<td>dazdemon</td>
<td>00:00:59</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2nd</td>
<td>WastedHarry</td>
<td>00:00:41</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>2nd</td>
<td>Ascla</td>
<td>00:00:37</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2nd</td>
<td>Antidote4Losing</td>
<td>00:00:41</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>2nd</td>
<td>BlackKnight9</td>
<td>00:00:48</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Given:

- Match outcomes: Orderings among k teams consisting of n1, n2, ..., nk players, respectively.

Questions:

- Skill ratings for each player such that
- Global ranking among all players
- Fair matches between teams of players

The Skill Rating Problem
**The Skill Rating Problem**

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

- **Questions:**
  - Skills $s_i$ for each player such that
  - Global ranking among all players
  - Fair matches between teams of players

### Table 1: Team Rankings

<table>
<thead>
<tr>
<th>Team</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Red Team</td>
<td>50</td>
</tr>
<tr>
<td>2nd Blue Team</td>
<td>40</td>
</tr>
</tbody>
</table>

### Table 2: Level and Gamertag Statistics

<table>
<thead>
<tr>
<th>Level</th>
<th>Gamertag</th>
<th>Avg. Life</th>
<th>Best Spree</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>BlueBot</td>
<td>00:00:49</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>1st</td>
<td>SniperEye</td>
<td>00:00:41</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>1st</td>
<td>ProThepirate</td>
<td>00:01:07</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>1st</td>
<td>dazdemon</td>
<td>00:00:59</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2nd</td>
<td>WastedHarry</td>
<td>00:00:41</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>2nd</td>
<td>Ascla</td>
<td>00:00:37</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2nd</td>
<td>Antidote4Losing</td>
<td>00:00:41</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>2nd</td>
<td>Blackknight9</td>
<td>00:00:48</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
**Given:**
- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

**Questions:**
- Skill ratings $s_i$ for each player such that
- Global ranking among all players
- Fair matches between teams of players

<table>
<thead>
<tr>
<th>Level</th>
<th>Gamertag</th>
<th>Avg. Life</th>
<th>Best Spree</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>N/A  SniperEye</td>
<td>N/A</td>
<td>N/A</td>
<td>25</td>
</tr>
<tr>
<td>2nd</td>
<td>N/A  xXxHALOxXx</td>
<td>N/A</td>
<td>N/A</td>
<td>24</td>
</tr>
<tr>
<td>3rd</td>
<td>N/A  AjaySandhu</td>
<td>N/A</td>
<td>N/A</td>
<td>15</td>
</tr>
<tr>
<td>3rd</td>
<td>N/A  AjaySandhu(G)</td>
<td>N/A</td>
<td>N/A</td>
<td>15</td>
</tr>
<tr>
<td>5th</td>
<td>N/A  Robert115</td>
<td>N/A</td>
<td>N/A</td>
<td>11</td>
</tr>
<tr>
<td>5th</td>
<td>N/A  TurboNegro84(G)</td>
<td>N/A</td>
<td>N/A</td>
<td>11</td>
</tr>
<tr>
<td>7th</td>
<td>N/A  TurboNegro84</td>
<td>N/A</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>8th</td>
<td>N/A  SniperEye(G)</td>
<td>N/A</td>
<td>N/A</td>
<td>1</td>
</tr>
</tbody>
</table>
Given:
- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

Questions:
- Skill ratings $s_i$ for each player such that
- Global ranking among all players
- Fair matches between teams of players
Given:
- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

Questions:
- Skill ratings $s_i$ for each player such that
- Global ranking among all players
- Fair matches between teams of players

The Skill Rating Problem
**Given:**

- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

**Questions:**

- Skill $s_i$ for each player such that

$$s_i > s_j \Rightarrow P(\text{Player } i \text{ wins}) > P(\text{Player } j \text{ wins})$$

- Global ranking among all players
- Fair matches between teams of players
Given:
- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

Questions:
- Skill $s_i$ for each player such that $P(\text{Player } i \text{ wins}) > P(\text{Player } j \text{ wins})$
Given:
- Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively.

Questions:
- Skill $s_i$ for each player such that
- Global ranking among all players
- Fair matches between teams of players

The Skill Rating Problem
• **Given:**
  – Match outcomes: Orderings among $k$ teams consisting of $n_1$, $n_2$, ..., $n_k$ players, respectively

• **Questions:**
  – Skill $s_i$ for each player such that
    \[ s_i > s_j \Rightarrow P(\text{Player } i \text{ wins}) > P(\text{Player } j \text{ wins}) \]
  – Global ranking among all players
  – Fair matches between teams of players
TrueSkill Technology Transfer

- Halo 2 Beta Test: Jul 2004
- Research: Sep 2004
- Halo 2: Dec 2004
• Latent Gaussian performance model for fixed skills
Two Player Match Outcome Model

- Latent Gaussian performance model for fixed skills
Two Player Match Outcome Model

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

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

\[
P(y_{12} = (1, 2)|p_1, p_2) = \mathbb{I}(p_1 > p_2)
\]
Two Team Match Outcome Model

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

\[ \text{Skill of team} = s_1 + s_2 + s_3 + s_4 \]
Skill of a team is the sum of the skills of its members.

\[ P(t_1|s_1, s_2) = \mathcal{N}(t_1; s_1 + s_2, 2 \cdot \beta^2) \]
Skill of a team is the sum of the skills of its members

\[ P(t_1|s_1, s_2) = \mathcal{N}(t_1; s_1 + s_2, 2 \cdot \beta^2) \]
Multiple Team Match Outcome Model

• Possible outcomes: Permutations of the teams

\[ S_1 \quad S_2 \quad S_3 \quad S_4 \]
Multiple Team Match Outcome Model

- Possible outcomes: Permutations of the teams

\[s_1 \quad s_2 \quad s_3 \quad s_4\]
\[t_1 \quad t_2 \quad t_3\]
Multiple Team Match Outcome Model

- Possible outcomes: Permutations of the teams

\[ P(y|t_1, t_2, t_3) = \mathbb{I}(y = (i, j, k)) \text{ where } t_i > t_j > t_k \]
Multiple Team Match Outcome Model

- Possible outcomes: Permutations of the teams

\[ P(y|t_1, t_2, t_3) = \mathbb{I}(y = (i, j, k)) \text{ where } t_i > t_j > t_k \]
• But we are interested in the (Gaussian) posterior!

\[ P(s_i|y = (1, 2, 3)) = \mathcal{N}(s_i; \mu_i, \sigma_i^2) \]
• But we are interested in the (Gaussian) posterior!

\[ P(s_i | y = (1, 2, 3)) = \mathcal{N}(s_i; \mu_i, \sigma_i^2) \]
Applications to Online Gaming

- **Leaderboard**
  - Global ranking of all players
Applications to Online Gaming

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

\[ \mu_i - 3 \cdot \sigma_i \]
Applications to Online Gaming

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

\[ \mu_i - 3 \cdot \sigma_i \]

- **Matchmaking**
Applications to Online Gaming

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

\[ \mu_i - 3 \cdot \sigma_i \]

• **Matchmaking**
Applications to Online Gaming

- **Leaderboard**
  - Global ranking of all players
  \[ \mu_i = 3 \cdot \sigma_i \]

- **Matchmaking**
Applications to Online Gaming

- **Leaderboard**
  - Global ranking of all players
  
  \[ \mu_i - 3 \cdot \sigma_i \]

- **Matchmaking**
  - For gamers: Most uncertain outcome
  - For inference: Most informative
  - Both are equivalent!

\[
P(p_i \approx p_j | \mu_i - \mu_j, \sigma_i^2 + \sigma_j^2) \]
\[
\frac{P(p_i \approx p_j | \mu_i - \mu_j = 0, \sigma_i^2 + \sigma_j^2 = 0)}
\]
• 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

Level

Number of Games

- char (Halo 2 rank)
- SQLWildman (Halo 2 rank)
Convergence Speed

Level vs Number of Games

- red: char (Halo 2 rank)
- blue: SQLWildman (Halo 2 rank)
Convergence Speed

- char (TrueSkill™)
- SQLWildman (TrueSkill™)
- char (Halo 2 rank)
- SQLWildman (Halo 2 rank)
Convergence Speed

Level vs. Number of Games for char (TrueSkill™), SQLWildman (TrueSkill™), char (Halo 2 rank), and SQLWildman (Halo 2 rank).
Convergence Speed

![Graph showing convergence speed with Level on the y-axis and Number of Games on the x-axis. The graph compares char (TrueSkill™) and SQLWildman (TrueSkill™) with Halo 2 rank metrics.](image)
Convergence Speed (ctd.)

- Winning probability
- Number of games played

- red: char wins
- blue: SQLWildman wins
- green: Both players draw

Graph shows the convergence speed of winning probabilities over the number of games played.
Convergence Speed (ctd.)

Number of games played

Winning probability

- red: char wins
- blue: SQLWildman wins
- green: Both players draw

5/8 games won by char
TrueSkill Technology Transfer

Halo 2 Beta Test
Reasech
Jul 2004
Sep 2004
Dec 2004
TrueSkill Technology Transfer

- Halo 2 Beta Test: Jul 2004
- Research: Sep 2004
- Code Devel: Dec 2004
- Code Devel: Mar 2005

Halo 2 Beta Test
TrueSkill Technology Transfer
• **Definition:** Graphical representation of joint probability distribution (Pearl, 1988)
  
  – Nodes: $\bigcirc = \text{Variables}$
  
  – Directed Edges: Conditional probability distribution
Directed Models: Bayesian Networks

• **Definition:** Graphical representation of joint probability distribution (Pearl, 1988)
  – Nodes: \( \bigcirc \) = Variables
  – Directed Edges: Conditional probability distribution

• **Semantic:**

\[
p(x) = \prod_i p(x_i | x_{\text{parents}(i)})
\]

– Ancestral relationship of dependency
Directed Models: Bayesian Networks

• **Definition:** Graphical representation of joint probability distribution (Pearl, 1988)
  - Nodes: \( \bigcirc \) = Variables
  - Directed Edges: Conditional probability distribution

• **Semantic:**
  \[
p(x) = \prod_i p(x_i | x_{\text{parents}(i)})
  \]
  - Ancestral relationship of dependency
  \[
p(a, b, c) = p(a) \cdot p(b) \cdot p(c | a, b)
  \]
**Definition:** Graphical representation of product structure of a function (Wiberg, 1996)

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

• **Semantic:**

\[ p(x) = \prod_{f} f(x_{V(f)}) \]

- Local variable dependency of factors
• **Definition:** Graphical representation of product structure of a function (Wiberg, 1996)
  
  - Nodes: ⬜ = Factors ⬜ = Variables
  - Edges: Dependencies of factors on variables.

• **Semantic:**

\[
p(x) = \prod_{f} f(x_{V(f)})
\]

- 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
Bayes’ law

\[ p(s|y) \propto p(y|s) \cdot p(s) \]
• **Bayes’ law**
  \[ p(s|y) \propto p(y|s) \cdot p(s) \]

• **Factorising prior**
  \[ p(s) = p(s_1) \cdot p(s_2) \]
• **Bayes’ law**

\[ p(s|y) \propto p(y|s) \cdot p(s) \]

• **Factorising prior**

\[ p(s) = p(s_1) \cdot p(s_2) \]
Factor Graphs and Bayes’ Law

- **Bayes’ law**
  \[ p(s|y) \propto p(y|s) \cdot p(s) \]

- **Factorising prior**
  \[ p(s) = p(s_1) \cdot p(s_2) \]

- **Factorising likelihood**
  \[ p(y, t, d|s) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d) \]
- **Bayes’ law**
  \[ p(s|y) \propto p(y|s) \cdot p(s) \]

- **Factorising prior**
  \[ p(s) = p(s_1) \cdot p(s_2) \]

- **Factorising likelihood**
  \[ p(y, t, d|s) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d) \]
• **Bayes’ law**
  \[ p(s|y) \propto p(y|s) \cdot p(s) \]

• **Factorising prior**
  \[ p(s) = p(s_1) \cdot p(s_2) \]

• **Factorising likelihood**
  \[ p(y, t, d|s) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d) \]

• **Inference:** Sum out latent variables
Factor Graphs and Bayes’ Law

- **Bayes’ law**
  \[ p(s|y) \propto p(y|s) \cdot p(s) \]

- **Factorising prior**
  \[ p(s) = p(s_1) \cdot p(s_2) \]

- **Factorising likelihood**
  \[ p(y, t, d|s) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d) \]

- **Inference: Sum out latent variables**
  \[ p(y|s) = \sum_t \sum_d p(y, t, d|s) \]
Factor Trees: Separation

\[ f_1(v, w) f_2(w, x) f_3(x, y) f_4(x, z) \]
Factor Trees: Separation

\[ p(w) = \sum_v \sum_x \sum_y \sum_z f_1(v, w)f_2(w, x)f_3(x, y)f_4(x, z) \]
Factor Trees: Separation

\[ p(w) = \left[ \sum_v f_1(v, w) \right] \cdot \left[ \sum_x \sum_y \sum_z f_2(w, x) f_3(x, y) f_4(x, z) \right] \]
Factor Trees: Separation

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

\[
p(w) = \left[ \sum_v f_1(v, w) \right] \cdot \left[ \sum_x \sum_y \sum_z f_2(w, x) f_3(x, y) f_4(x, z) \right]
\]
Factor Trees: Separation

Observation: Sum of products becomes product of sums of all messages from neighbouring factors to variable!
Messages: From Factors To Variables

\[ m_{f_2 \rightarrow w} (w) = \sum_x \sum_y \sum_z f_2(w, x) f_3(x, y) f_4(x, z) \]
Messages: From Factors To Variables

\[ m_{f_2 \rightarrow w}(w) = \sum_x f_2(w, x) \cdot \left[ \sum_y \sum_z f_3(x, y) f_4(x, z) \right] \]
Observation: Factors only need to sum out all their local variables!
Observation: Factors only need to sum out all their local variables!
Messages: From Variables To Factors

\[ m_{x \rightarrow f_2}(x) = \sum_y \sum_z f_3(x, y) f_4(x, z) \]
Messages: From Variables To Factors

\[ m_{x\rightarrow f_2}(x) = \left[ \sum_y f_3(x, y) \right] \cdot \left[ \sum_z f_4(x, z) \right] \]
**Observation:** Variables pass on the product of all incoming messages!

$$m_{x\rightarrow f_2(x)} = \left[ \sum_y f_3(x, y) \right] \cdot \left[ \sum_z f_4(x, z) \right]$$
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)

\[
p(t) = \prod_{f \in F_t} m_{f \rightarrow t}(t)
\]

\[
m_{f \rightarrow t_1}(t_1) = \sum_{t_2} \sum_{t_3} \cdots \sum_{t_n} f(t_1, t_2, t_3, \ldots) \prod_{i>1} m_{t_i \rightarrow f}(t_i)
\]

\[
m_{t \rightarrow f}(t) = \prod_{f_j \in F_t \setminus \{f\}} m_{f_j \rightarrow t}(t)
\]

- 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!
Approximate Message Passing

\[ \tilde{m}_{t \to f}(t) \quad \ast \quad m_{f \to t}(t) \]
Approximate Message Passing

\[ \tilde{m}_{t \rightarrow f}(t) \quad \ast \quad m_{f \rightarrow t}(t) \quad = \quad p(t) \]
Approximate Message Passing

\( \tilde{m}_{t \rightarrow f}(t) \times m_{f \rightarrow t}(t) = p(t) \approx \hat{p}(t) \)
Approximate Message Passing

\[ \hat{m}_{t \rightarrow f}(t) \ast m_{f \rightarrow t}(t) = p(t) \]

\[ \hat{m}_{t \rightarrow f}(t) \ast \hat{m}_{f \rightarrow t}(t) = \hat{p}(t) \]
Approximate Message Passing

\[ \tilde{m}_{t \to f}(t) \ast m_{f \to t}(t) = p(t) \]

\[ \tilde{m}_{t \to f}(t) \ast \tilde{m}_{f \to t}(t) = \hat{p}(t) \]
Efficient Approximate Inference
Efficient Approximate Inference

Gaussian Prior Factors

$s_1$  $s_2$  $s_3$  $s_4$

$t_1$  $t_2$  $t_3$
Efficient Approximate Inference

Gaussian Prior Factors

Ranking Likelihood Factors

$s_1$, $s_2$, $s_3$, $s_4$

$t_1$, $t_2$, $t_3$
Fast and efficient approximate message passing using Expectation Propagation
Efficient Approximate Inference
Efficient Approximate Inference
Efficient Approximate Inference

Diagram with nodes labeled as $s_1$, $s_2$, $s_3$, $s_4$, $t_1$, $t_2$, and $t_3$. The nodes are connected by directed edges.
Efficient Approximate Inference
Efficient Approximate Inference
Efficient Approximate Inference
TrueSkill Technology Transfer

Halo 2 Beta Test
Jul 2004

Research
Sep 2004

Code Devel
Dec 2004

Work with Game Devs
Mar 2005

Research
Nov 2005

Research
Dec 2005
• **Code size:** 1400 LOC + 1400 LOC

• **Project size:** 2 project / 21 files

• **Development time:** 2 month

• **Features**
  
  – Parser: High performance (> 2GB logs in 1 hour)
  
  – Parser: Recreation of matchmaking server status
  
  – Viewer: SQL database integration (deep schema)
Xbox Live Activity viewer

- **Code size**: 1400 LOC + 1400 LOC
- **Project size**: 2 project / 21 files
- **Development time**: 2 month
- **Features**
  - Parser: High performance (> 2GB logs in 1 hour)
  - Parser: Recreation of matchmaking server status
  - Viewer: SQL database integration (deep schema)
• Code size: 1400 LOC + 1400 LOC
• Project size: 2 project / 21 files
• Development time: 2 month

• Features
  – Parser: High performance (> 2GB logs in 1 hour)
  – Parser: Recreation of matchmaking server status
  – Viewer: SQL database integration (deep schema)
Skill Distributions of Online Games

Golf (18 holes): 60 levels

Car racing (3-4 laps): 40 levels

UNO (chance game): 10 levels
TrueSkill Technology Transfer

- Beta Test: Jul 2004
- Halo 2: Sep 2004
- Research: Dec 2004
- Code Devel: Mar 2005
- Work with Game Devs: Nov 2005
- Research: Dec 2005
- Develop: Mar 2006
- Work with Game Devs: Jun 2006
TrueSkill Technology Transfer

- **Jul 2004**: Halo 2 Beta Test
- **Sep 2004**: Research
- **Dec 2004**: Code Devel
- **Mar 2005**: Work with Game Devs
- **Nov 2005**: Research
- **Dec 2005**: Develop
- **Mar 2006**: Work with Game Devs
- **Jun 2006**: Simulation work and Tool Development for Halo 3
- **Mar 2007**: Research

- Halo 2
- Beta Test
- Research
- Code Devel
- Work with Game Devs
- Research
- Develop
- Work with Game Devs
- Simulation work and Tool Development for Halo 3
• **Questions**
  – Controllable player skill progression (slow-down!)
  – Controllable skill distributions (re-ordering)

• **Simulations**
  – Large scale simulation of > 8,000,000,000 matches
  – Distributed application written in C# using .Net remoting

• **Tools**
  – Result viewer (Logged results: 52 GB of data)
  – Real-time simulator of partial update
Halo 3 Simulation Result Viewer

- **Code size:** 1800 LOC
- **Project size:** 11 files
- **Development time:** 2 month

**Features**
- Multithreaded histogram viewer (due to file size)
- Real-time spline editor (monotonically increasing)
- Based on WinForms (compatibility)
Halo 3 Simulation Result Viewer

- **Code size**: 1800 LOC
- **Project size**: 11 files
- **Development time**: 2 months
- **Features**
  - Multithreaded histogram viewer (due to file size)
  - Real-time spline editor (monotonically increasing)
  - Based on WinForms (compatibility)
Halo 3 Simulation Result Viewer

- Code size: 1800 LOC
- Project size: 11 files
- Development time: 2 months
- Features:
  - Multithreaded histogram viewer (due to file size)
  - Real-time spline editor (monotonically increasing)
  - Based on WinForms (compatibility)

TrueSkill™ is a ranking and matchmaking system for Xbox 360 online gaming. The skill of each gamer is internally represented by two numbers called Mu and Sigma.

- Mu is the mean skill of the player and represents the current best guess of the system.
- Sigma is a measure of uncertainty or spread of the player's skill.

These numbers are set to Mu=3.0 and Sigma=1.0 initially before a gamer has played a single game in a particular game mode. After resolving a game outcome in the form of a ranking of participating players (including draws) TrueSkill updates both numbers, Mu and Sigma, for all the participating players.

- Generally the Mu values of the winner(s) increase and the Mu values of the loser(s) decrease.
- Sigma values generally decrease after each game unless the game outcome was fully expected by the system.

The TrueSkill of a player can be calculated as TrueSkill = 25/2 (Mu - 3 * Sigma) and represents a conservative estimate of a player's skill. We also use the term Uncertainty
TrueSkill Technology Transfer

- Jul 2004: Halo 2 Beta Test
- Sep 2004: Research
- Dec 2004: Code Devel
- Mar 2005: Work with Game Devs
- Nov 2005: Research
- Dec 2005: Develop
- Mar 2006: Work with Game Devs
- Jun 2006: Simulation work and Tool Development for Halo 3
- Mar 2007: Research
- May 2007:

- Halo 2 Beta Test
- Research
- Code Devel
- Work with Game Devs
- Research
- Develop
- Work with Game Devs
- Simulation work and Tool Development for Halo 3
- Research
TrueSkill Technology Transfer

- **Jul 2004**: Beta Test
- **Sep 2004**: Halo 2
- **Dec 2004**: Research
- **Mar 2005**: Code Devel
- **Nov 2005**: Work with Game Devs
- **Dec 2005**: Research
- **Mar 2006**: Develop
- **Jun 2006**: Work with Game Devs
- **Mar 2007**: Simulation work and Tool Development for Halo 3
- **May 2007**: Research
- **Jul 2007**: Work with Bungie
- **Code size:** 2600 LOC
- **Project size:** 10 files
- **Development time:** 1 month

**Features**
- SQL database integration (analysis of beta test data)
- Full integration of C# TrueSkill code (.Net library)
- Real time changes
Halo 3 Partial Update Analyser

- **Code size:** 2600 LOC

- **Project size:** 10 files

- **Development time:** 1 month

- **Features**
  - SQL database integration (analysis of beta test data)
  - Full integration of C# TrueSkill code (.Net library)
  - Real time changes

Censored

Halo 3 Delta Partial Update Analyser 3.0
Halo 3 Partial Update Analyser

- **Code size:** 2600 LOC
- **Project size:** 10 files
- **Development time:** 1 month

**Features**
- SQL database integration (analysis of beta test data)
- Full integration of C# TrueSkill code (.Net library)
- Real time changes
Halo 3 Public Beta Analysis

1 games played

Skill Level

Winning probability (in %)
**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)
TrueSkill Technology Transfer

- Beta Test: Jul 2004
- Halo 2: Sep 2004
- Research: Dec 2004
- Code Devel: Mar 2005
- Work with Game Devs: Nov 2005
- Research: Dec 2005
- Develop: Mar 2006
- Work with Game Devs: Jun 2006
- Research: Mar 2007
- Work with Bungie: May 2007
- Tool Develop (Halo 3): Jul 2007
- Tool Develop (Halo 3): Nov 2007

- Simulation work and Tool Development for Halo 3: Mar 2007

- Halo 2 Beta Test: Sep 2004

- Halo 2: Beta Test
Lessons Learned:

1. Pure research takes a short amount of time
2. Most of development was tool development
3. A platform feature only lives with a community
4. Mathematical Optimality ≠ Fun Experience
Joint work with Thore Graepel, Joaquin Quiñonero Candela, Onno Zoeter, Tom Borchert, Phillip Trelford
AdPredictor Technology Transfer

Jan 2007
Problem Identify

Mar 2007
Why Predict Probability-of-Click?
Why Predict Probability-of-Click?
Why Predict Probability-of-Click?
Why Predict Probability-of-Click?

<table>
<thead>
<tr>
<th>Cost</th>
<th>Probability</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.00</td>
<td>10%</td>
<td>$0.10</td>
</tr>
<tr>
<td>$2.00</td>
<td>4%</td>
<td>$0.08</td>
</tr>
<tr>
<td>$0.10</td>
<td>50%</td>
<td>$0.05</td>
</tr>
</tbody>
</table>

Related searches:
- Seattle Weather
- Seattle Times
- Seattle Hotels
- Craigslist Seattle
- Seattle Washington
- Seattle Mariners
- Craigs List Seattle
- Seattle Seahawks

Sponsored sites
- Seattle Washington Rates
- Visiting Seattle & Need a Hotel?
- Seattle Washington Hotel Bargains!
Why Predict Probability-of-Click?

- 10% = $0.10
- 4% = $0.08
- 50% = $0.05
Why Predict Probability-of-Click?

- Display (according to expected revenue)
  \[ b_1 \cdot p_1 \geq b_2 \cdot p_2 \geq \ldots \]
Why Predict Probability-of-Click?

- **Display** (according to expected revenue)
  \[ b_1 \cdot p_1 \geq b_2 \cdot p_2 \geq \cdots \]

- **Charge** (per click)
  \[ c_i = b_{i+1} \cdot \frac{p_{i+1}}{p_i} \]
Why Predict Probability-of-Click?

- **Display** (according to expected revenue)
  \[ b_1 \cdot p_1 \geq b_2 \cdot p_2 \geq \cdots \]

- **Charge** (per click)
  \[ c_i = b_{i+1} \cdot \frac{p_{i+1}}{p_i} \]

- **Advantages of improved probability estimates:**
  - Increase user satisfaction by better targeting
  - Fairer charges to advertisers
  - Increase revenue by showing ads with high click-thru rate
AdPredictor Technology Transfer

Problem Identify

Jan 2007

Mar 2007
AdPredictor Technology Transfer

Problem Identify
AdCenter Compete

Jan 2007
Mar 2007
May 2007
User interaction
User interaction → Raw Logs → Structured Data
User interaction → Raw Logs → Structured Data

• Why structured data?
  – Data validation and cleaning
  – Principled feature transformations
• **Code size:** 500 LOC
• **Project size:** 1 file
• **Development time:** 2 weeks

• **Features**
  - Code defines the schema (unlike LINQ)!
  - High-performance insertion via computed bulk-insertion with automated key propagation
  - Code sample is now part of the F# distribution
/// A single page-view

type PageView =
{
    ClientDateTime : DateTime
    GmtSeconds : int
    TargetDomainId : int16
    Medium : MediumType option
    StartPosition : int
    PageNum : byte
    [<SqlStringLengthAttribute(256)>]
    Query : string
    Gender : Gender option
    AgeBucket : AgeGroup option
    ReturnedAdCnt : byte
    AbTestingType : byte option
    AlgorithmId : int option
    ANID : int128 option
    GUID : int128 option
    [<SqlStringLengthAttribute(15)>]
    PassportZipCode : string option
    [<SqlStringLengthAttribute(2)>]
    PassportCountry : string option
    PassportRegion : int
    [<SqlStringLengthAttribute(2)>]
    PassportOccupation : char
    LocationCountry : int
    LocationState : int
    LocationMetroArea : int
    CategoryId : int16
    SubCategoryId : int16
    FormCode : int16
    ReturnedAds : Advertisement array
}

/// Different types of media

type MediumType =
| PaidSearch
| ContextualSearch

/// A single displayed advertisement

type Advertisement =
{
    AdId : int
    OrderItemID : int
    CampDayId : int16
    CampHourNum : byte
    ProductId : ProductType
    MatchType : MatchType
    AdLayoutId : AdLayout
    RelativePosition : byte
    DeliveryEngineRank : int16
    ActualBid : int
    ProbabilityOfClick : int16
    MatchScore : int
    ImpressionCnt : int
    ClickCnt : int
    ConversionCnt : int
    TotalCost : int
}
/// A single page-view

type PageView =
{
    ClientDateTime : DateTime
    GmtSeconds     : int
    TargetDomainId : int16
    Medium         : MediumType option
    StartPosition  : int
    PageNum        : byte
    [<SqlStringLengthAttribute (256)>]
    Query          : string
    Gender         : Gender option
    AgeBucket      : AgeGroup option
    ReturnedAdCnt  : byte
    AbTestingType  : byte option
    AlgorithmId    : int option
    ANID           : int128 option
    GUID           : int128 option
    [<SqlStringLengthAttribute (15)>]
    PassportZipCode: string option
    [<SqlStringLengthAttribute (2)>]
    PassportCountry: string option
    [<SqlStringLengthAttribute (2)>]
    PassportRegion : int
    PassportOccupation : char
    LocationCountry : int
    LocationState  : int
    LocationMetroArea : int
    CategoryId     : int16
    SubCategoryId  : int16
    FormCode       : int16
    ReturnedAds    : Advertisement array
}
Uncertainty: Bayesian Probabilities

Client IP

102.34.12.201
15.70.165.9
221.98.2.187
92.154.3.86

Match Type

Exact Match
Broad Match

Position

ML-1
SB-1
SB-2

p(pClick)
Training Algorithm in Action

\[ w_1 + z + w_2 \]
Training Algorithm in Action

- Prediction
Training Algorithm in Action

No Click

Click

Prediction
Training Algorithm in Action

No Click

Click

+ Prediction

Training/Update

Prediction

Training/Update
\[ \mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \]
\[ \sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2 \]

\[ h(t) = \frac{N(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
\[ \mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \]

\[ \sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2 \]

\[ h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
Inference: An Optimization View

\[ \mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \]

\[ \sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2 \]

\[ h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
\[ \mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \]

\[ \sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2 \]

\[ h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
Inference: An Optimization View

\[ \mu_i \leftarrow \mu_i + \frac{\sigma^2_i}{s} \cdot h \left( \frac{\sum_{j=1}^{d} \mu_j}{s} \right) \]

\[ \sigma^2_i \leftarrow \sigma^2_i \left( 1 - \frac{\sigma^2_i}{s^2} \cdot g \left( \frac{\sum_{j=1}^{d} \mu_j}{s} \right) \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma^2_j \]

\[ h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
\[
\mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h
\]

\[
\sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \right)
\]

\[
s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2
\]

\[
h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)}
\]

\[
g(t) = h(t) \cdot [h(t) + t]
\]
Inference: An Optimization View

\[ \mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \]

\[ \sigma_i^2 \leftarrow \sigma_i^2 \left( 1 - \frac{\sigma_i^2}{s^2} \cdot g \left[ \frac{\sum_{j=1}^{d} \mu_j}{s} \right] \right) \]

\[ s^2 = \beta^2 + \sum_{j=1}^{d} \sigma_j^2 \]

\[ h(t) = \frac{\mathcal{N}(t; 0, 1)}{\Phi(t)} \]

\[ g(t) = h(t) \cdot [h(t) + t] \]
AdPredictor Technology Transfer

Problem Identify

AdCenter Compete

Jan 2007

Mar 2007

May 2007
AdPredictor Technology Transfer

Problem Identification
Jan 2007

AdCenter Compete
Mar 2007

Offline Evaluation
May 2007

Aug 2007
Client IP; Mean & Variance

ClientIP Parameters

σ²

μ

Graph showing the distribution of Client IP parameters with mean and variance.
Client IP: Mean & Variance

ClientIP Parameters

Low clickers
AdPredictor Technology Transfer

Problem Identify | AdCenter Compete | Offline Evaluation

Mar 2007
May 2007
Aug 2007

Problem Identify
AdCenter Compete
Offline Evaluation

AdCenter
Compete
Offline Evaluation
AdPredictor Technology Transfer

- Problem Identify
- AdCenter Compete
- Offline Evaluation
- Working with Development Team on scalable Training

- Jan 2007
- Mar 2007
- May 2007
- Aug 2007
- Mar 2008
AdPredictor Technology Transfer

Problem Identify
Jan 2007

AdCenter Compete
Mar 2007

Offline Evaluation
May 2007

Working with Development Team on scalable Training
Aug 2007

Research Analysis
Mar 2008

Research Analysis
Jun 2008
AdPredictor Technology Transfer

- Problem Identify
- AdCenter Compete
- Offline Evaluation
- Working with Development Team on scalable Training
- Research Analysis
- Development of Tools

Distributed Conditional Models

\[ \theta \]

\[ Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7 \]
Distributed Conditional Models
Distributed Conditional Models
Distributed Conditional Models
Distributed Conditional Models

Data Messages ("Compute")
Distributed Conditional Models

\[ \theta \]

Data Messages ("Compute")
Distributed Conditional Models

Data Messages ("Compute")
Distributed Conditional Models

Belief Store ("Memory")

Data Messages ("Compute")
Distributed Conditional Models

Belief Store ("Memory")

Message Passing ("Communicate")

Data Messages ("Compute")
Relation to Map-Reduce

- Map-Reduce
Map-Reduce

\[ p(\theta|x, y) \propto \prod_k f_k(Y_k|\theta, X_k) \cdot p(\theta) \]
• **Map-Reduce**
  
  - **Map**: Data nodes compute messages $m_{F_k \to \mu}$ from data $y_i$ and $m_{\mu \to F_k}$
• **Map-Reduce**
  - **Map**: Data nodes compute messages $m_{F_k \rightarrow \mu}$ from data $y_i$ and $m_{\mu \rightarrow F_k}$
  - **Reduce**: Combine messages $m_{F_k \rightarrow \mu}$ into $p_{\mu}$ by multiplication

\[
p(\theta | x, y) \propto \prod_{k} f_k(Y_k | \theta, X_k) \cdot p(\theta)
\]
• Map-Reduce
  – **Map**: Data nodes compute messages $m_{F_k \to \mu}$ from data $y_i$ and $m_{\mu \to F_k}$
  – **Reduce**: Combine messages $m_{F_k \to \mu}$ into $p_{\mu}$ by multiplication
  – Vanilla MR is a single pass only!
• Map-Reduce
  – **Map**: Data nodes compute messages $m_{F_k \rightarrow \mu}$ from data $y_i$ and $m_{\mu \rightarrow F_k}$
  – **Reduce**: Combine messages $m_{F_k \rightarrow \mu}$ into $p_{\mu}$ by multiplication
  – Vanilla MR is a single pass only!

• Caveats:
• **Map-Reduce**
  
  - **Map**: Data nodes compute messages \( m_{F_k \rightarrow \mu} \) from data \( y_i \) and \( m_{\mu \rightarrow F_k} \)
  
  - **Reduce**: Combine messages \( m_{F_k \rightarrow \mu} \) into \( p_{\mu} \) by multiplication
  
  - Vanilla MR is a single pass only!

• **Caveats**:
  
  - Approximate data factors need all incoming message \( m_{F_k \rightarrow \mu} \)!

\[
p(\theta | x, y) \propto \prod_k f_k(Y_k | \theta, X_k) \cdot p(\theta)
\]
• **Map-Reduce**
  - **Map**: Data nodes compute messages $m_{F_k \rightarrow \mu}$ from data $y_i$ and $m_{\mu \rightarrow F_k}$
  - **Reduce**: Combine messages $m_{F_k \rightarrow \mu}$ into $p_{\mu}$ by multiplication
  - Vanilla MR is a single pass only!

• **Caveats**:
  - Approximate data factors need all incoming message $m_{F_k \rightarrow \mu}$!
  - Each machine needs to be able to store the belief over $\mu$
AdPredictor Technology Transfer

- Problem Identify (Jan 2007)
- AdCenter Compete (Mar 2007)
- Offline Evaluation (May 2007)
- Working with Development Team on scalable Training (Aug 2007)
- Research Analysis (Mar 2008)
- Development of Tools (Jun 2008)
- Beta Test (Dec 2008)
- Beta Test (Mar 2009)
• **Lessons Learned:**

1. Pure research takes a short amount of time
2. Development takes much longer than planned
3. Metrics are important and part of the transfer
4. Develop for scale from Day 1
Technology Transfer in Numbers

- **TrueSkill**
  - Evangelisation: 18
  - Development: 7
  - Research: 3
  - Problem Identification: 3

- **AdPredictor**
  - Evangelisation: 6
  - Development: 12
  - Research: 7
  - Problem Identification: 3
Overview

- Background
- Technology Transfer Case Studies
  - TrueSkill: Gamer Rating and Matchmaking
  - Click-Through Rate Prediction in Online Advertising
- Technology Transfer Lessons
  - Process
  - Technical
Technology Transfer: Lessons

- Identify the problem
- Identify the customer
- Be the customer!

Problem vs. Technique

- Do not study the existing literature first!
- Don’t be afraid to be wrong

Try Out and Then Study

- Reduce risk!
- Understand the decision and engineering process
- Respect timelines

Under-Promise and Over-Deliver

- Write code!
- Work with developers – not executives!

Coding

- Simplify to its bare minimum.
- Develop tools and help adoption with customers

Simplicity

- Be in for the long haul – years is normal
- Do not aim for a quick win!

Long Run
• Technology Transfer is Highly Rewarding!
• Practical problems start new research directions!
• Graphical models are a very powerful language:
  – Modeling (Bayes Nets)
  – Algorithm development (Sum-Product)
  – Highly modular (Local Factors)
  – (Relatively) easy to teach (Pictorial)
• Lots of open problems in learning theory that considers system limitations (CPU, storage, etc.)