Technology Transfer of Machine Learning in Practice

Ralf Herbrich
Amazon

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

Background


1992 – 1997 (Berlin, Diploma)


1997 – 2000 (Berlin, PhD)


2000 – 2009 (Microsoft Research)


2009 – 2011 (Microsoft)


2011 – 2012 (Facebook)


2012 – Present (Amazon)

Technology Transfer

- **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?)

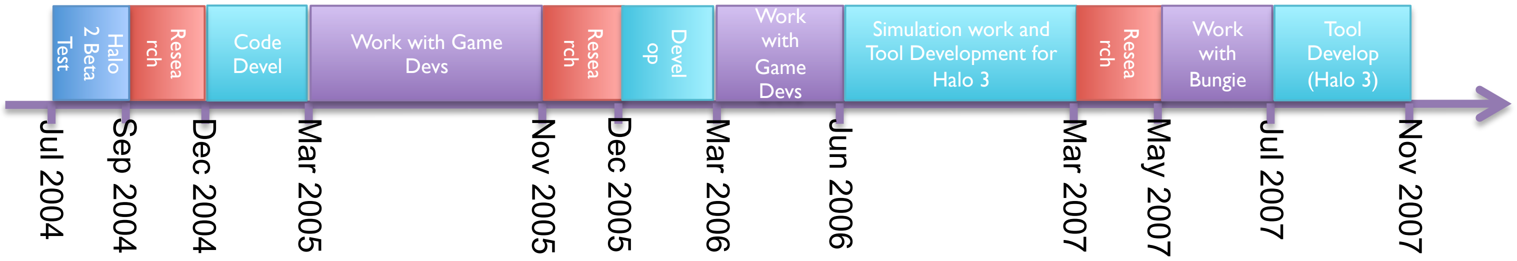
Overview

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

TrueSkill™

Joint work with Thore Graepel, Tom Minka & Phillip Trelford

TrueSkill Technology Transfer



- **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

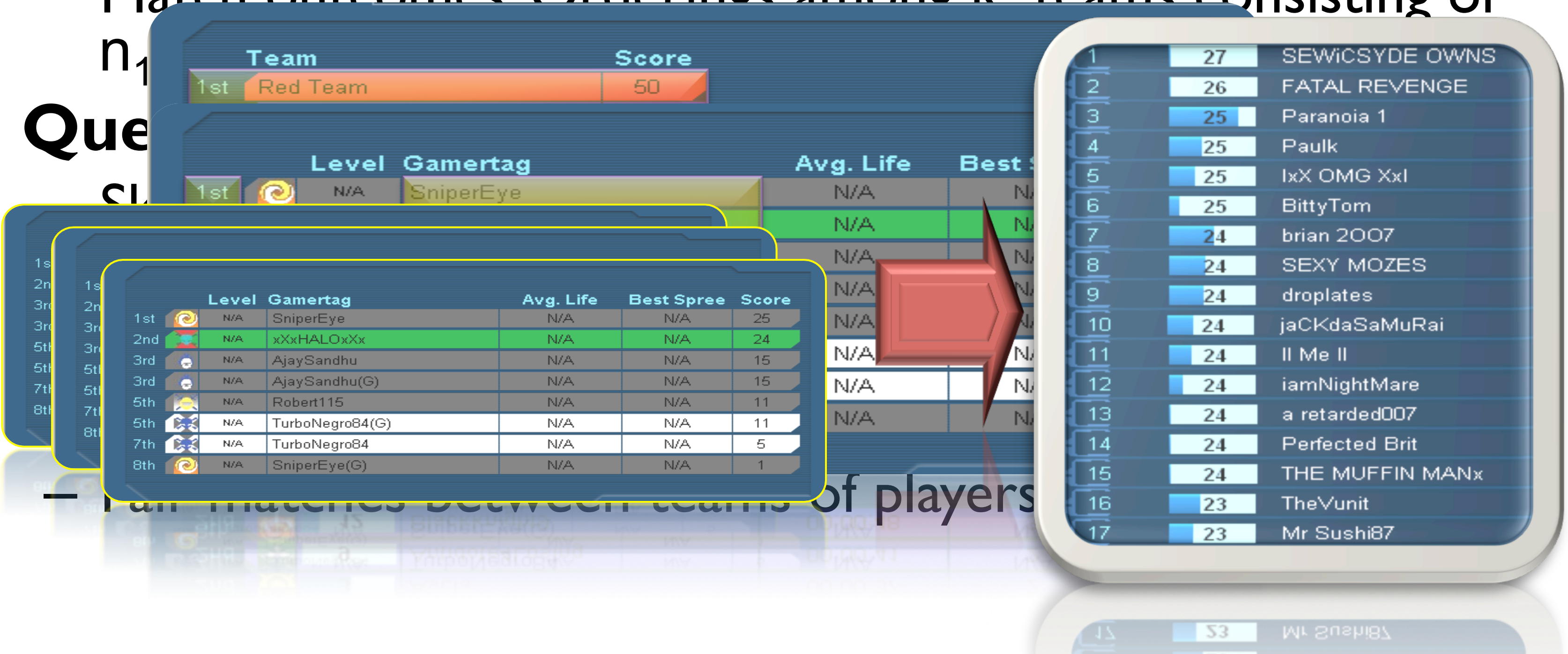
TrueSkill Technology Transfer

The Skill Rating Problem

- **Given:**
  - Match outcomes: Orderings among k teams consisting of n1
- **Que**

  Sk

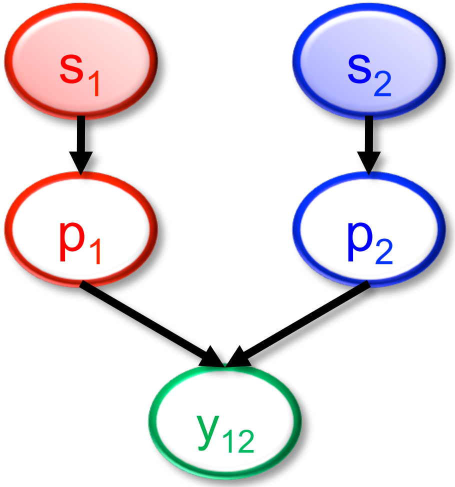
  – of players



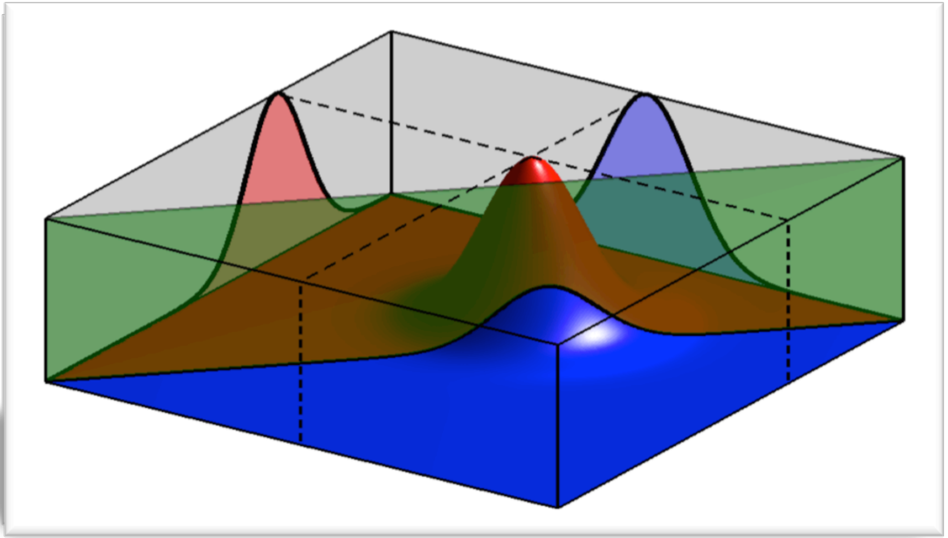
TrueSkill Technology Transfer

Two Player Match Outcome Model

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

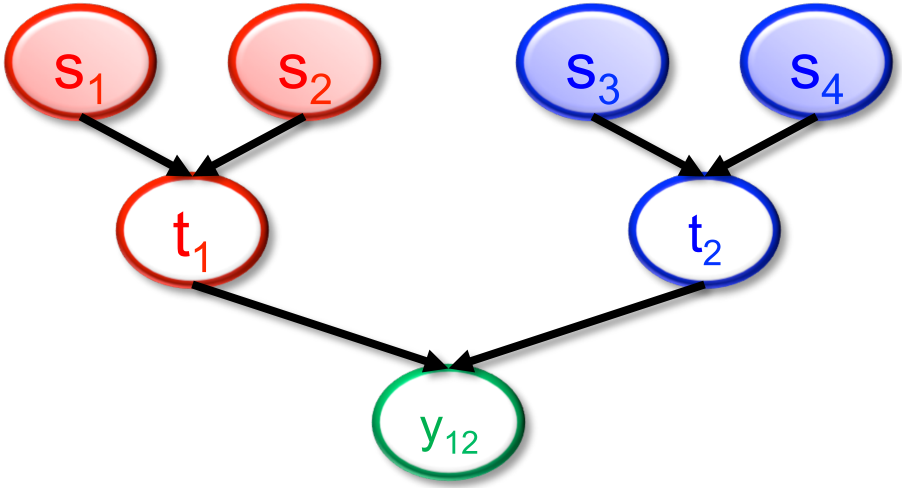


$$\mathbf{P}(y_{12} = (1,2) | p_1, p_2) = \mathbb{I}(p_1 > p_2)$$



Two Team Match Outcome Model

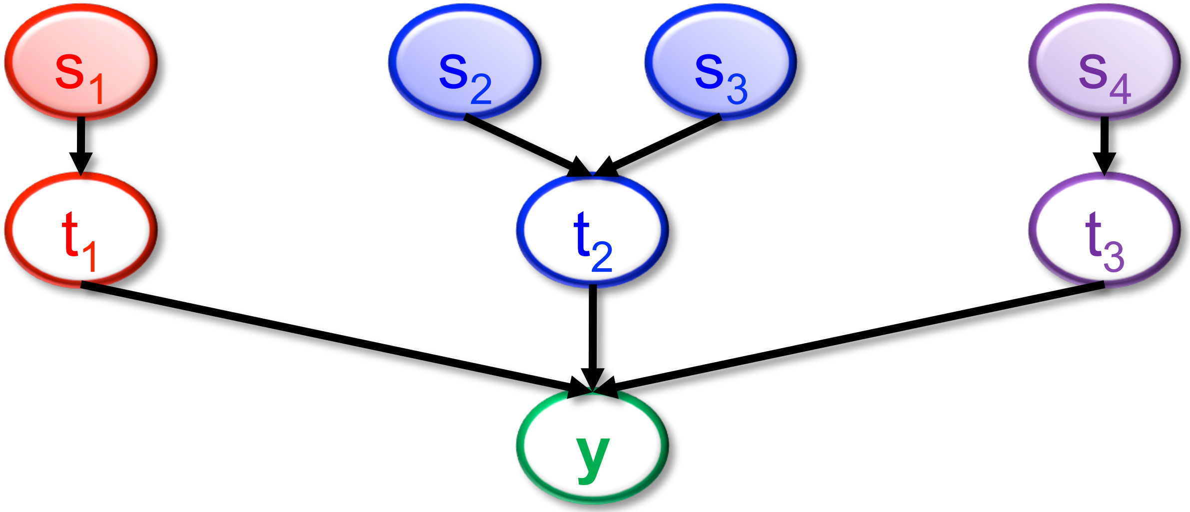
- Skill of a team is the sum of the skills of its members



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

Multiple Team Match Outcome Model

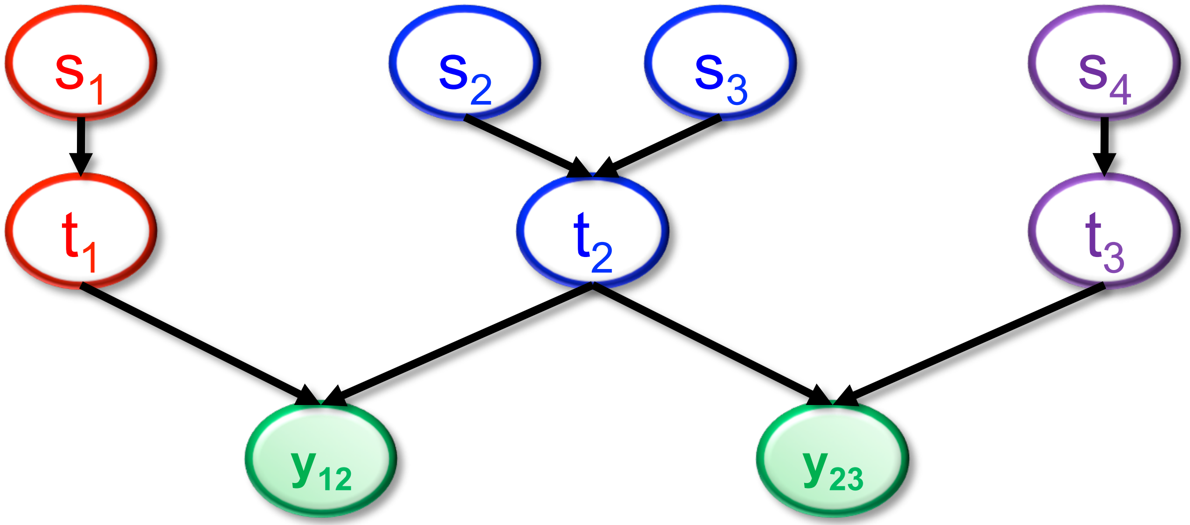
- Possible outcomes: Permutations of the teams



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

Multiple Team Match Outcome Model

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



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

Applications to Gaming

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

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

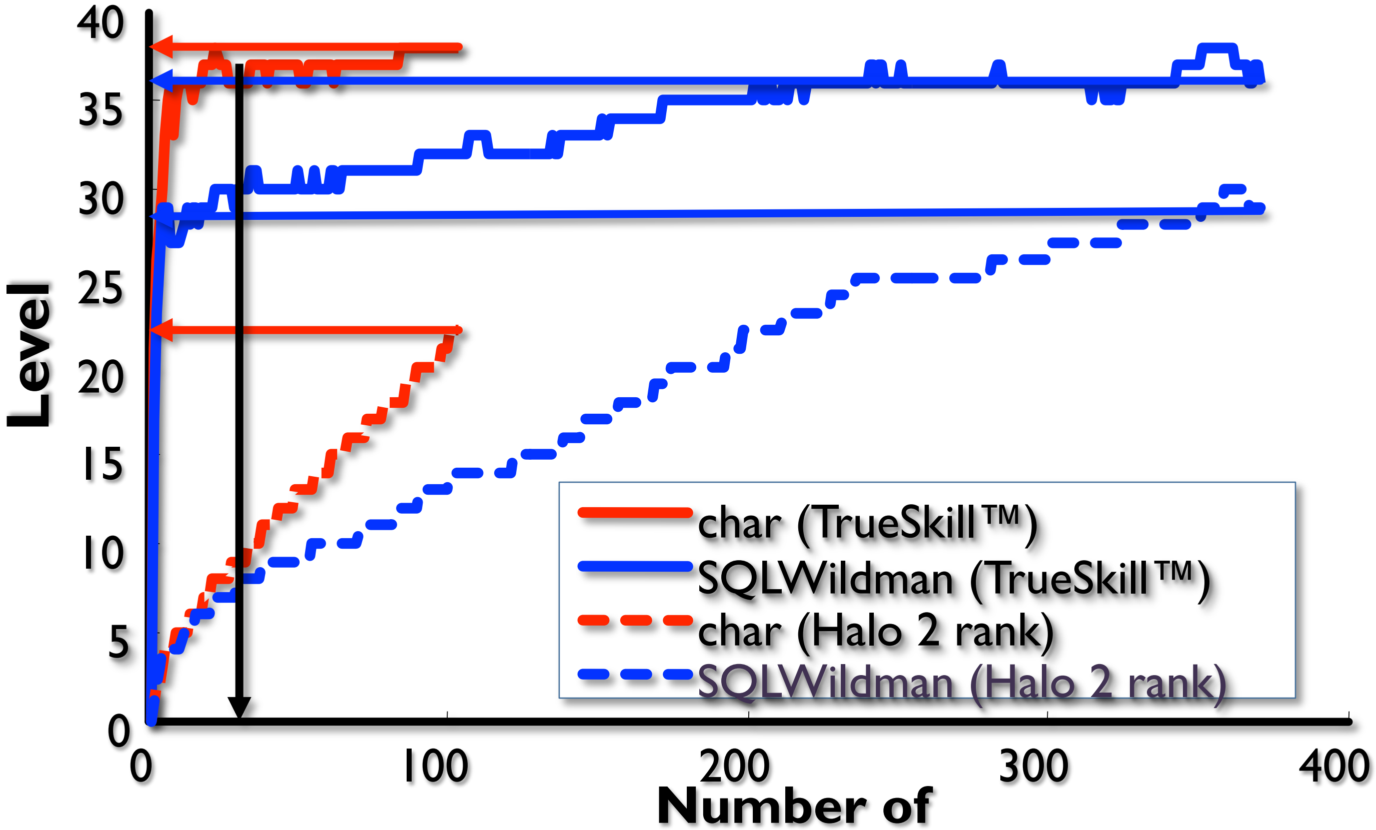
- **Matchmaking**
  - For gamers: Most uncertain o
  - For inference: Most informativ

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

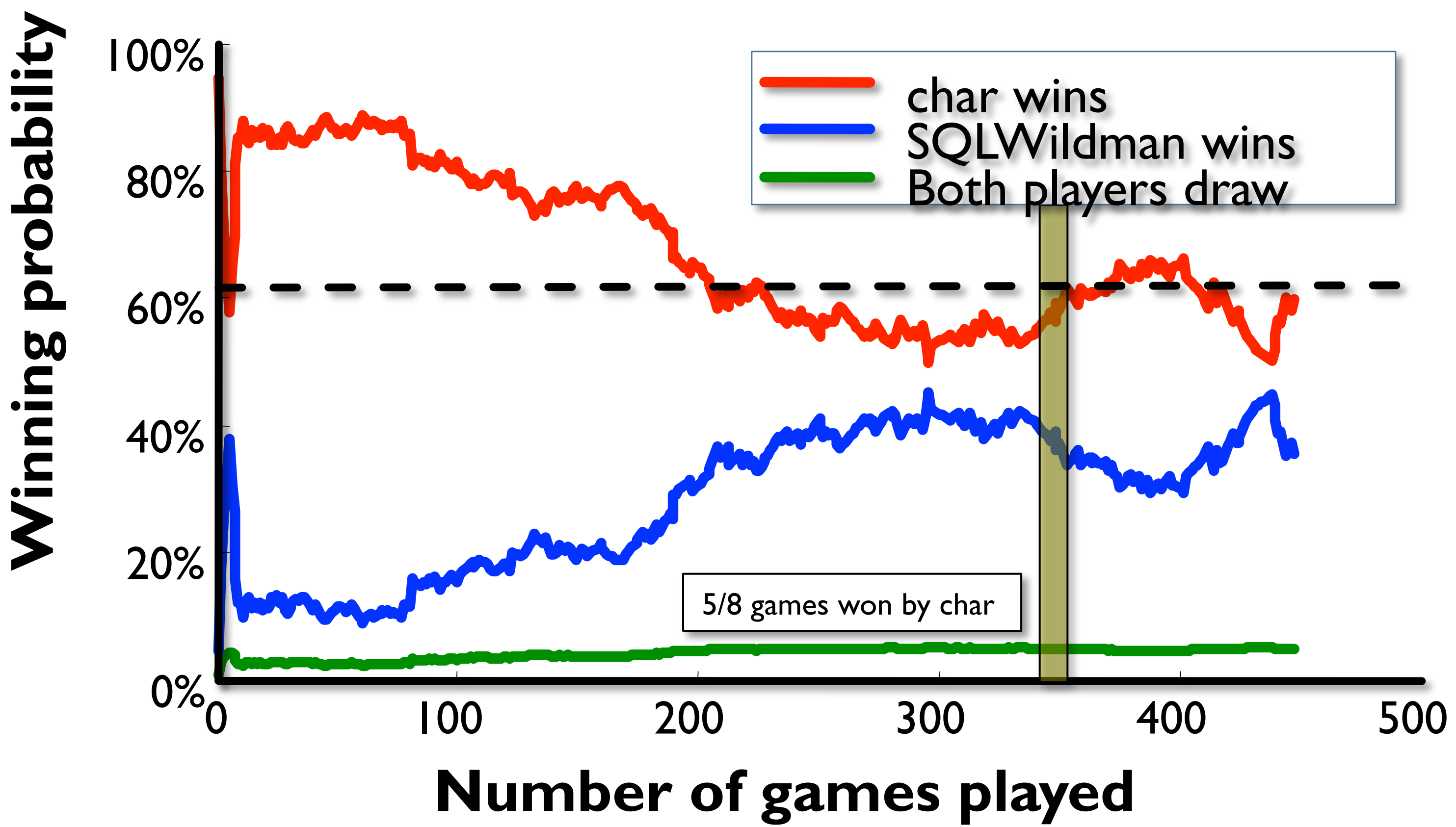
Experimental Setup

- **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

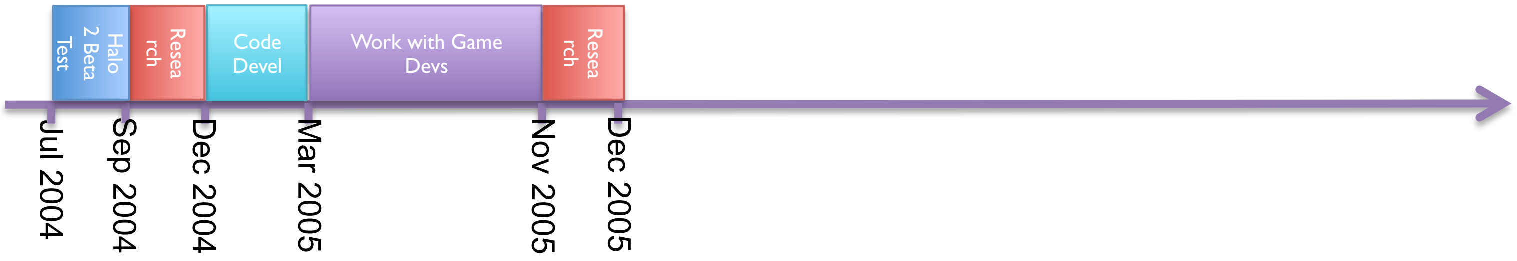
Convergence Speed



Convergence Speed (ctd.)



TrueSkill Technology Transfer



Graphical Models

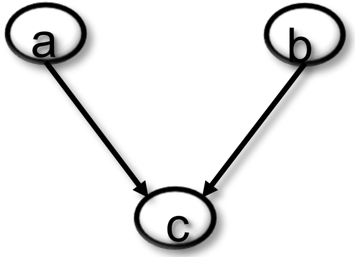
- **Definition:** Graphical representation of joint probability distribution
  - Nodes:  = Variables
  - Edges: Relationship between variables
- **Variables**:
  - Observed Variables: Data
  - Unobserved Variables: 'Causes' + Temporary/Latent
- **Key Questions**:
  - (Conditional) *Dependency*: $p(a,b|c)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

Directed Models: Bayesian Networks

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

- **Semantic:**

$$p(\mathbf{x}) = \prod_i p\left(x_i | \mathbf{x}_{\text{parents}(i)}\right)$$



  - Ancestral relationship of dependency

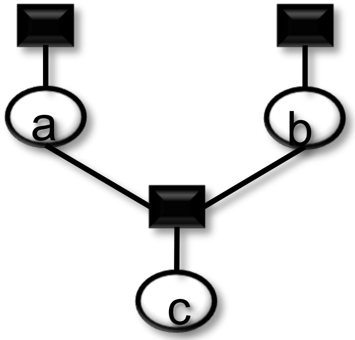
$$p(a, b, c) = p(a) \cdot p(b) \cdot p(c|a, b)$$

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

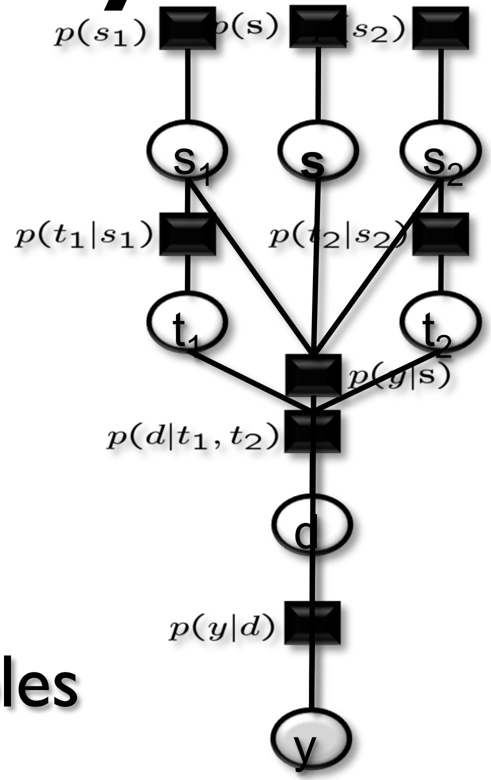
$$p(\mathbf{s}) = p(s_1) \cdot p(s_2)$$

- Factorising likelihood

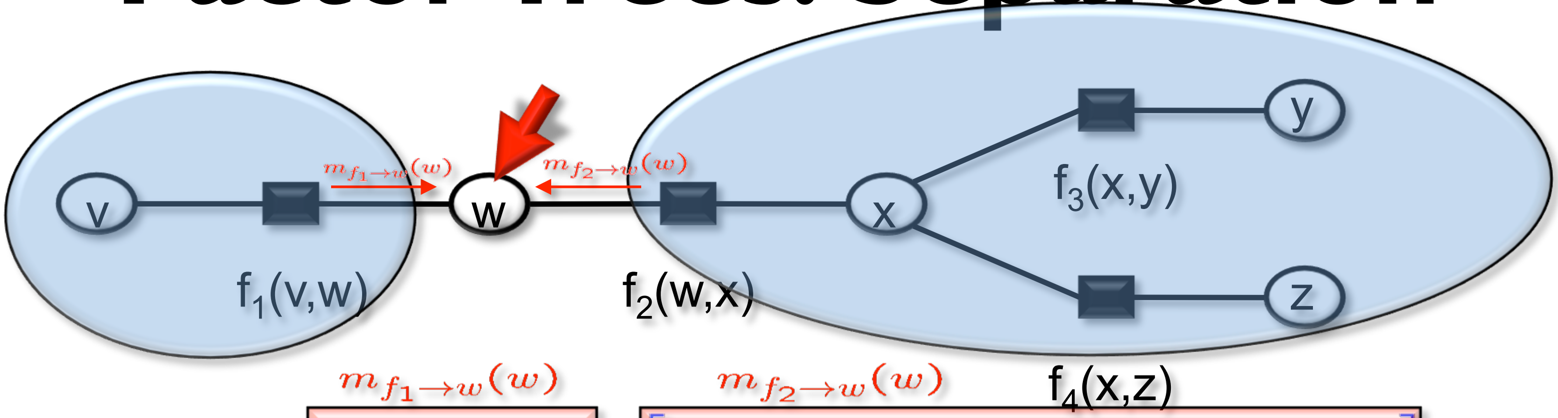
$$p(y, \mathbf{t}, d|\mathbf{s}) = \prod_i p(t_i|s_i) \cdot p(d|t_1, t_2) \cdot p(y|d)$$

- Inference: Sum out latent variables

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



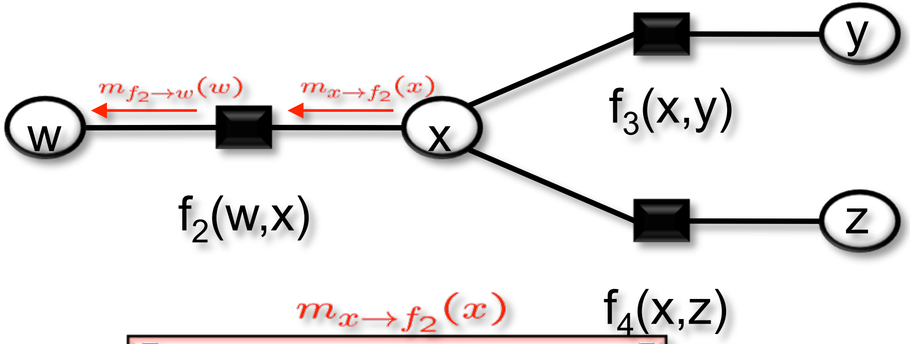
Factor Trees: Separation



$$p(w) = \overbrace{\left[\sum_v f_1(v,w)\right]}^{m_{f_1 \to w}(w)} \overbrace{\left[\sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z)\right]}^{m_{f_2 \to w}(w)}$$

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

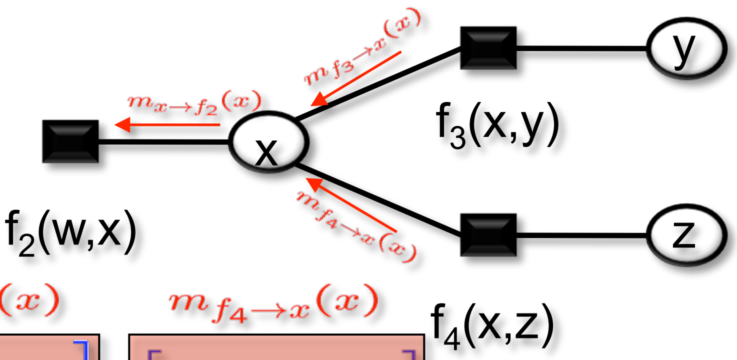
Messages: From Factors To Variables



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

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

Messages: From Variables To Factors



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

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

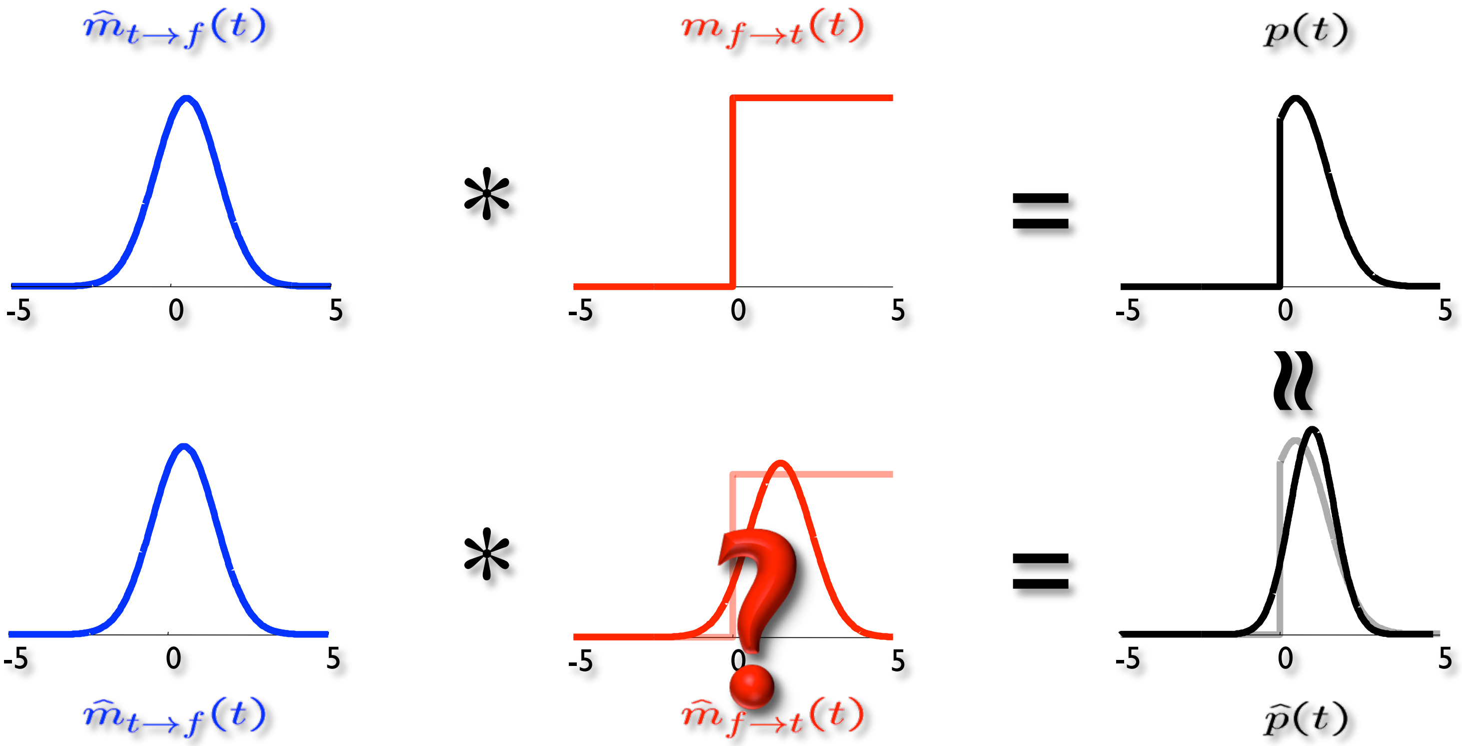
The Sum-Product Algorithm

- Three update equations (Aji & McEliece, 1997)

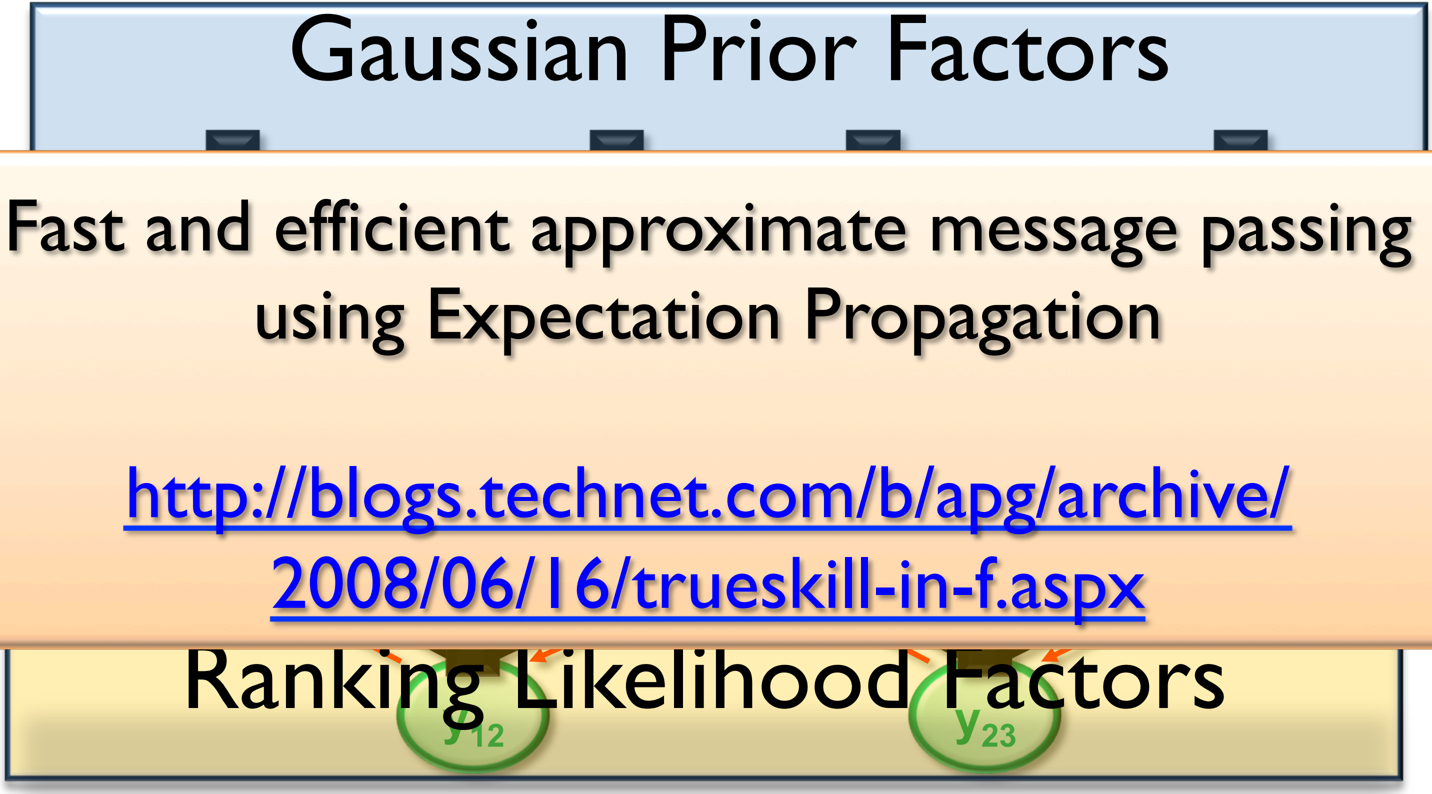
$$
\begin{aligned}
p(t) &= \prod_{f \in F_t} m_{f \to t}(t) \\
m_{f \to t_1}(t_1) &= \sum_{t_2} \sum_{t_3} \cdots \sum_{t_n} f(t_1, t_2, t_3, \ldots) \prod_{i>1} m_{t_i \to f}(t_i) \\
m_{t \to f}(t) &= \prod_{f_j \in F_t \setminus \{f\}} m_{f_j \to t}(t)
\end{aligned}
$$

- 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!

Approximate Message Passing



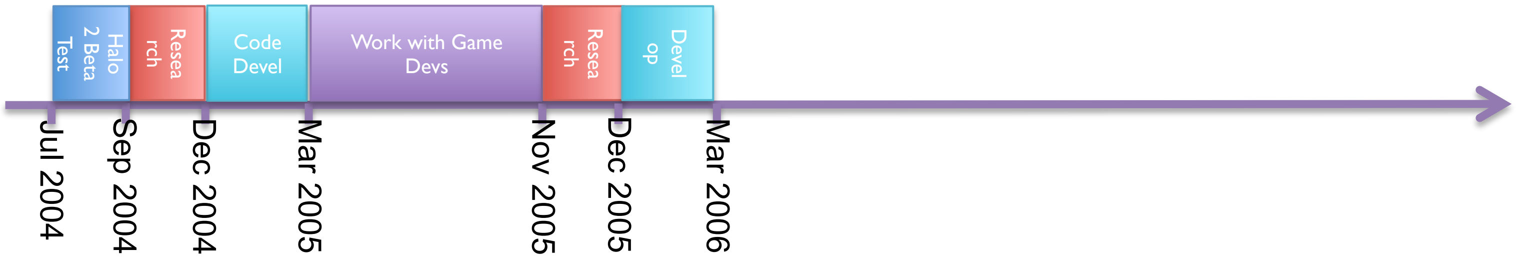
Efficient Approximate Inference



Fast and efficient approximate message passing using Expectation Propagation

http://blogs.technet.com/b/apg/archive/2008/06/16/trueskill-in-f.aspx

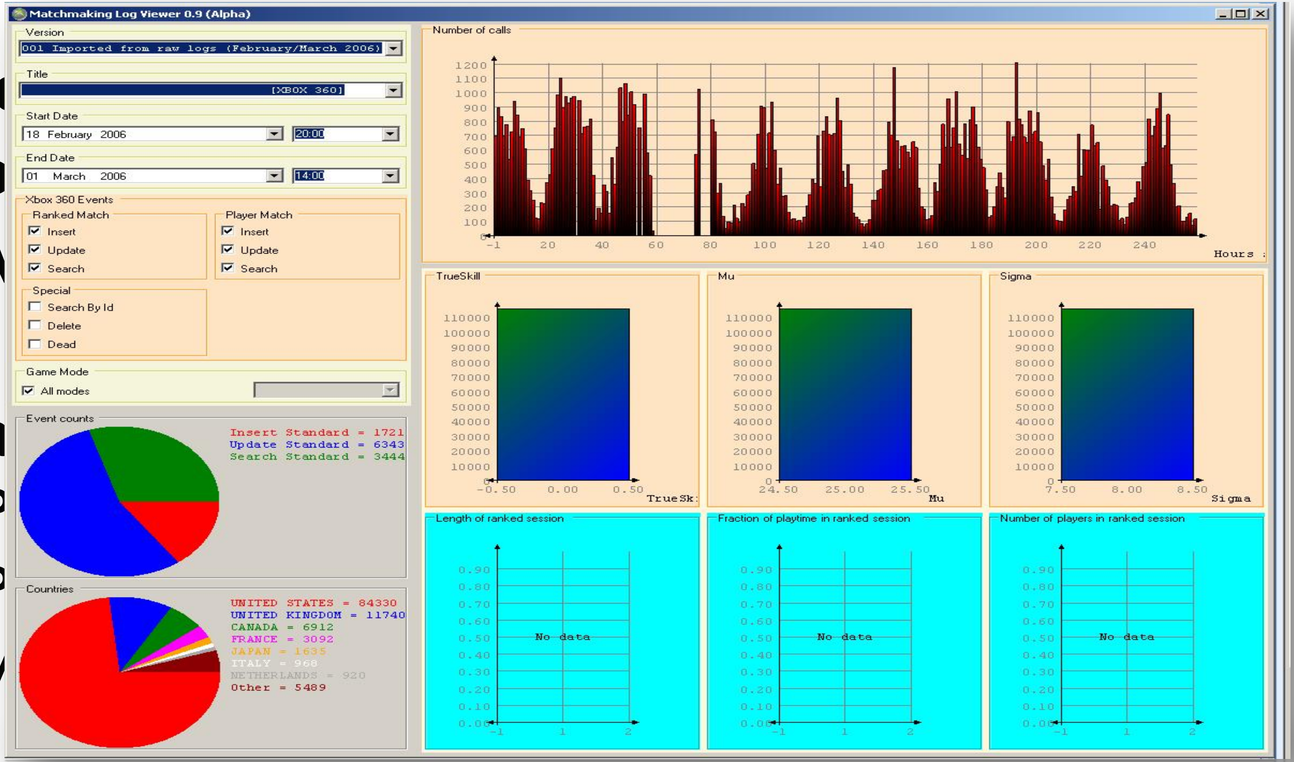
TrueSkill Technology Transfer



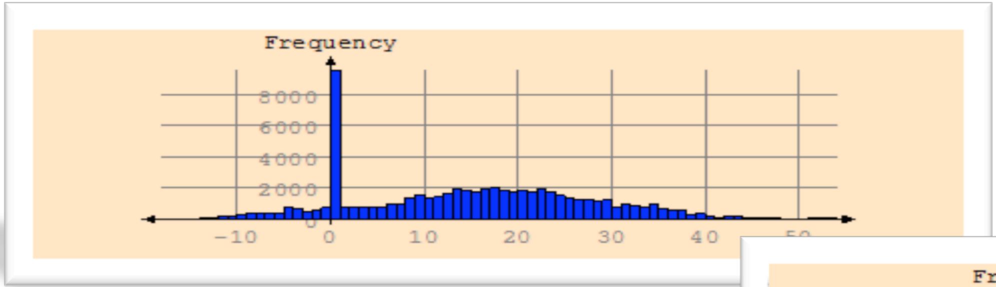
Xbox Live Activity viewer

- Co
- Pro
- Dev

- Fea
  - P
  - P
  - V

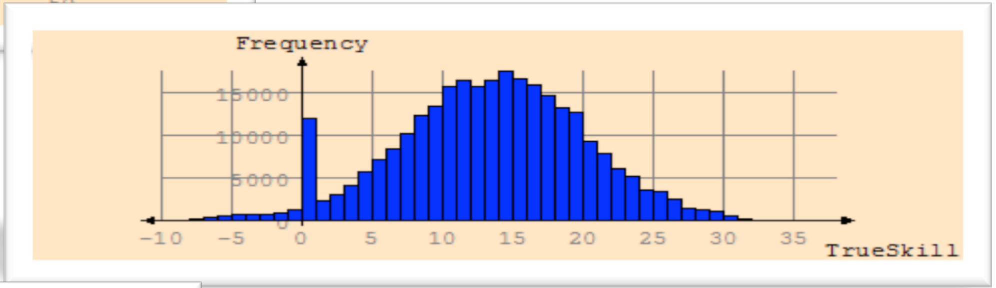


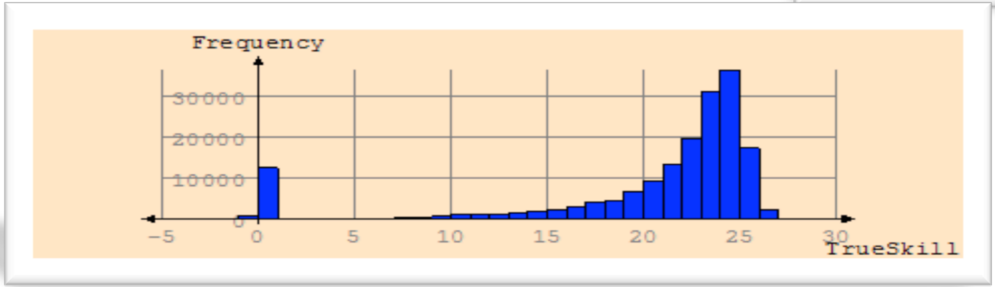
Skill Distributions of Online Games



Golf (18 holes): 60 levels

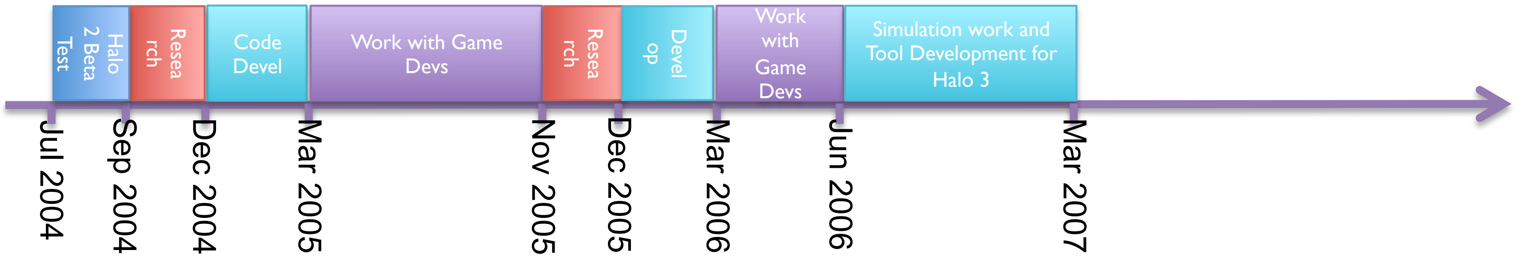
Car racing (3-4 laps): 40 levels



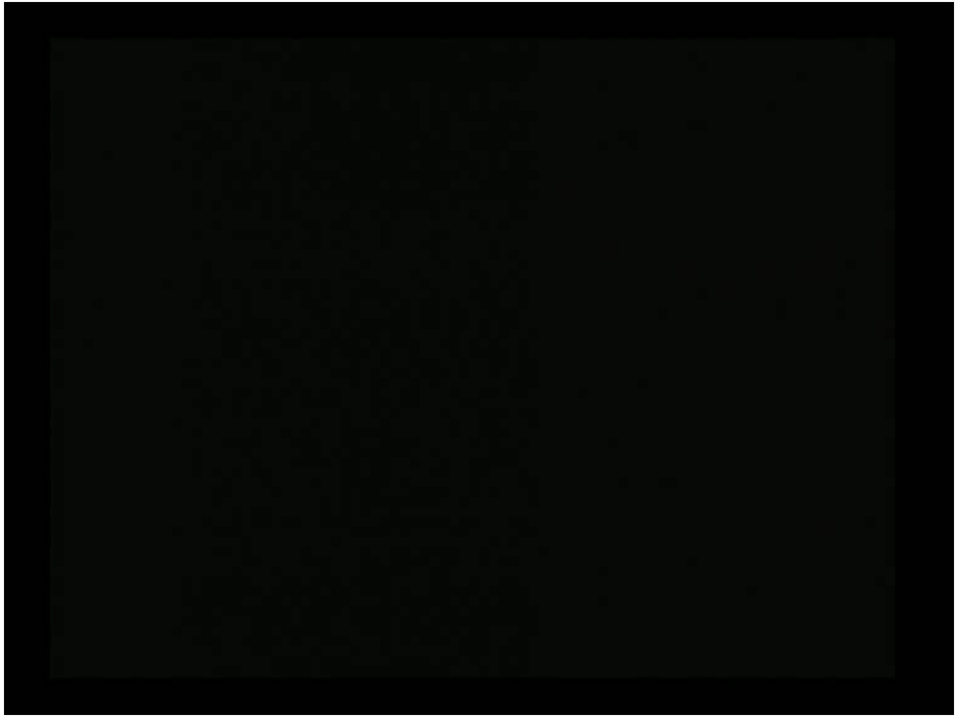


UNO (chance game): 10 levels

TrueSkill Technology Transfer



Halo 3 in Action



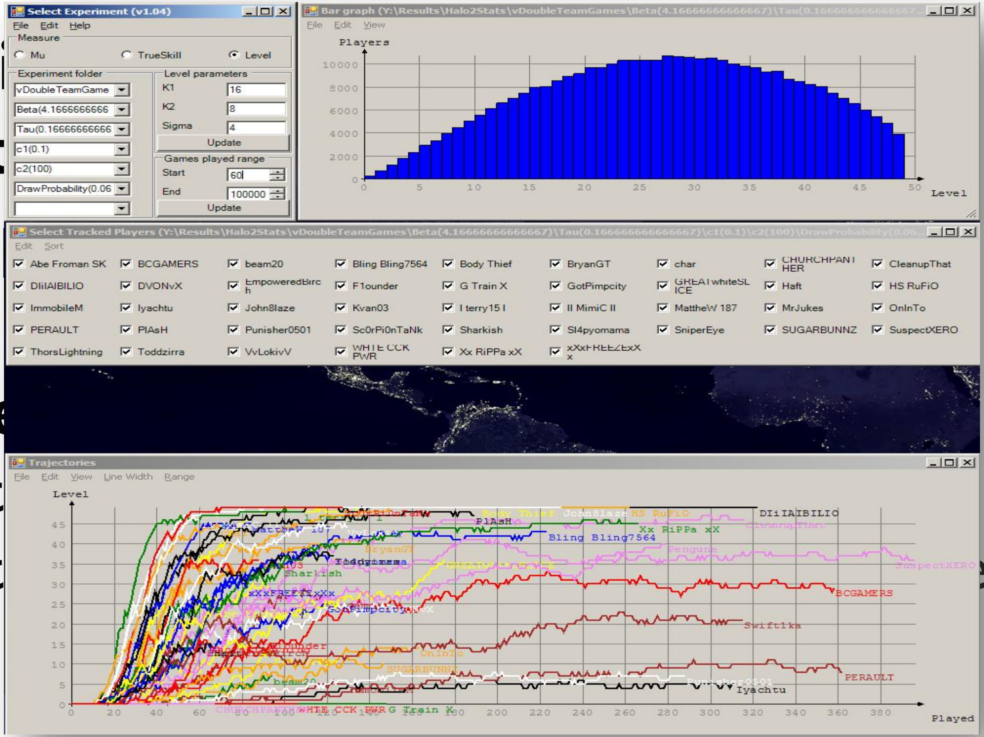
Tools for Halo 3

- **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

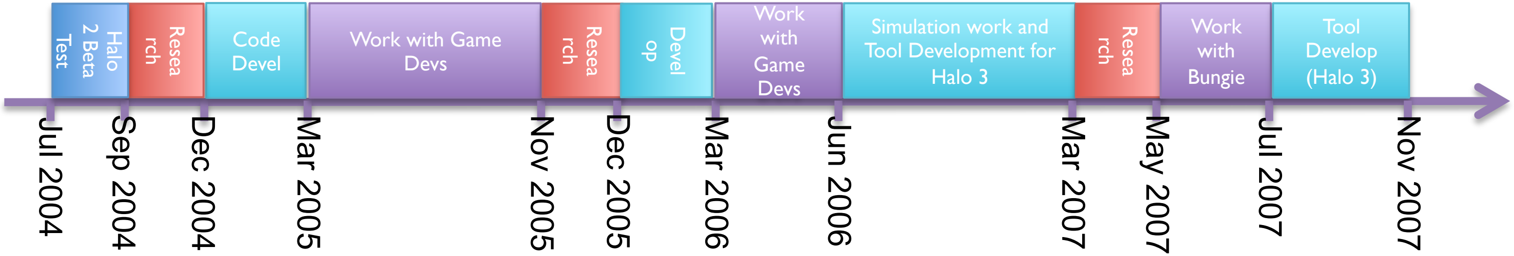
Halo 3 Simulation Result Viewer

- Code si
- Project
- Develo

- Feature
  - Multit size)
  - Real-t easing)
  - Based



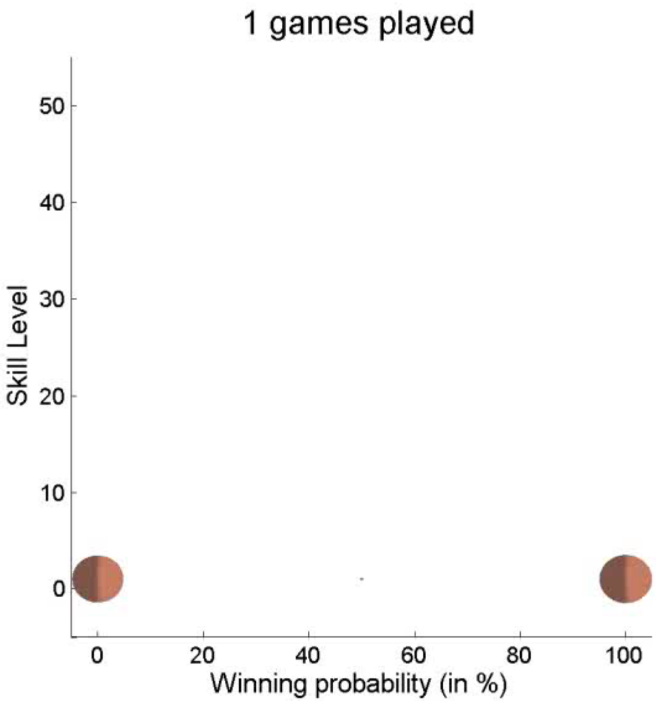
TrueSkill Technology Transfer



Halo 3 Partial Update Analyser

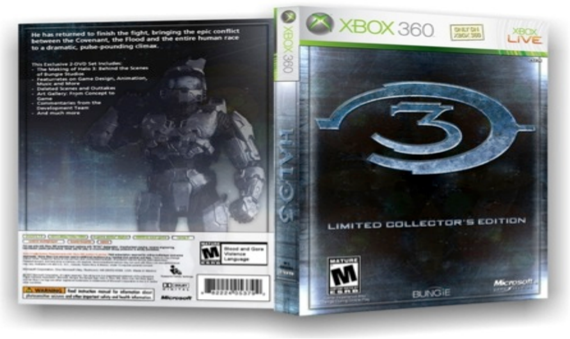
- **Code size:** 2600 LOC
- **P**
- **D**
- **F**
  - –
  - –
  - Real time changes

Halo 3 Public Beta Analysis

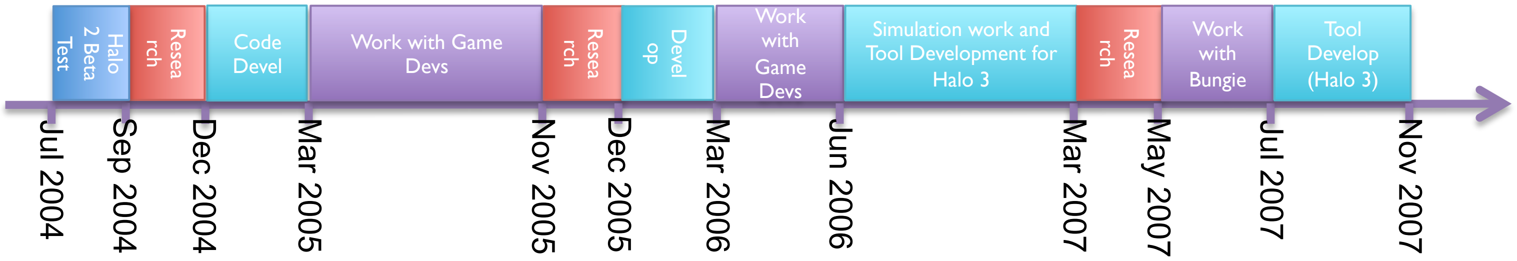


Xbox 360 & Halo 3

- **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)

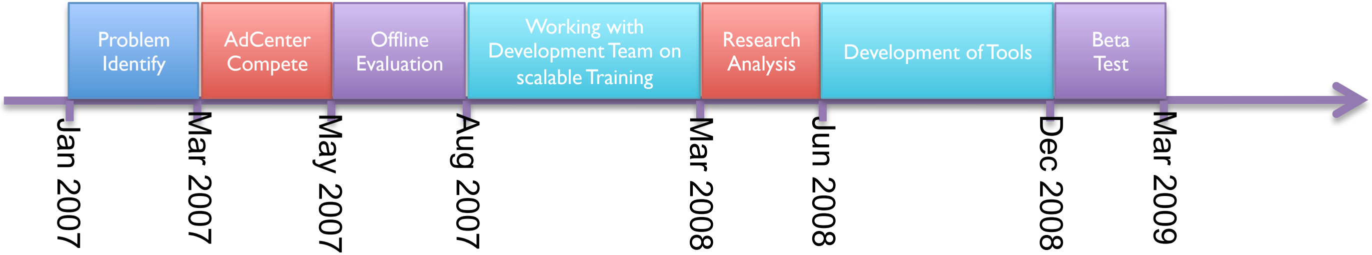


TrueSkill Technology Transfer



- **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

AdPredictor Technology Transfer



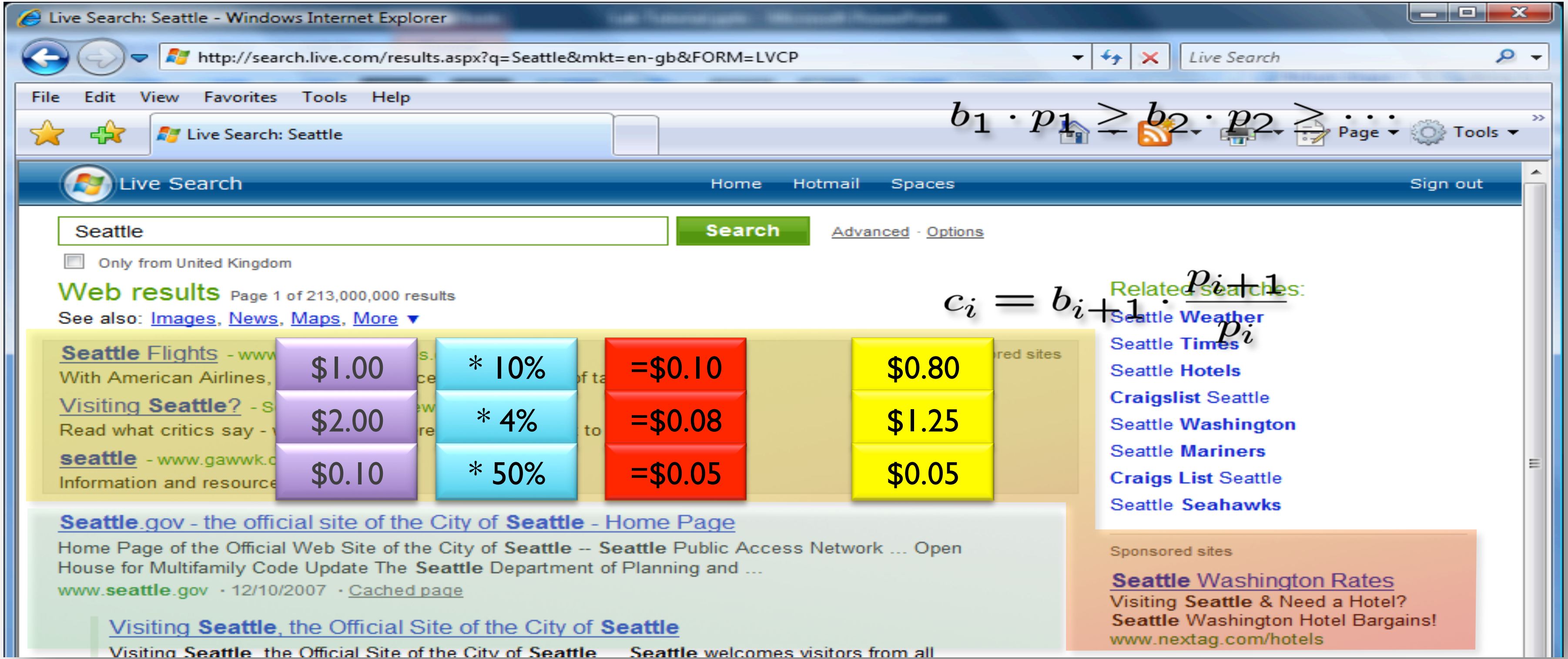
- **Lessons Learned:**
  1. Pure research takes a short amount of time
  2. Development takes much longer than planned
  3. Counter-factual analysis and metrics are important
  4. Develop for scale from Day 1

adPredictor

Joint work with Thore Graepel, Joaquin Quiñonero Candela, Onno Zoeter, Tom Borchert , Phillip Trelford

AdPredictor Technology Transfer

Why Predict Probability-of-Click?



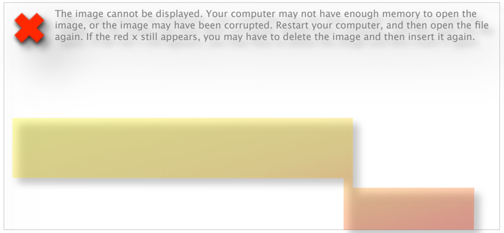
$$b_1 \cdot p_1 \geq b_2 \cdot p_2 \geq \cdots$$

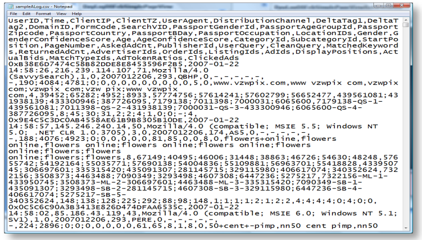
$$c_i = b_{i+1} \cdot \frac{p_{i+1}}{p_i}$$

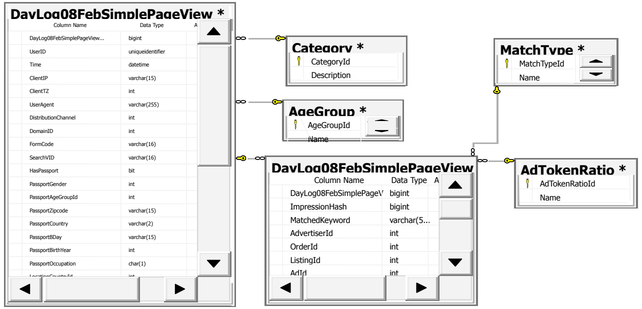
$1.00	* 10%	=$0.10	$0.80
$2.00	* 4%	=$0.08	$1.25
$0.10	* 50%	=$0.05	$0.05

AdPredictor Technology Transfer

The Flow of Information







User interaction → Raw Logs → Structured Data

- Why structured data?
  - Data validation and cleaning
  - Principled feature transformations

SQL Schema Generator

- **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

Strong Typing and SQL Datastores

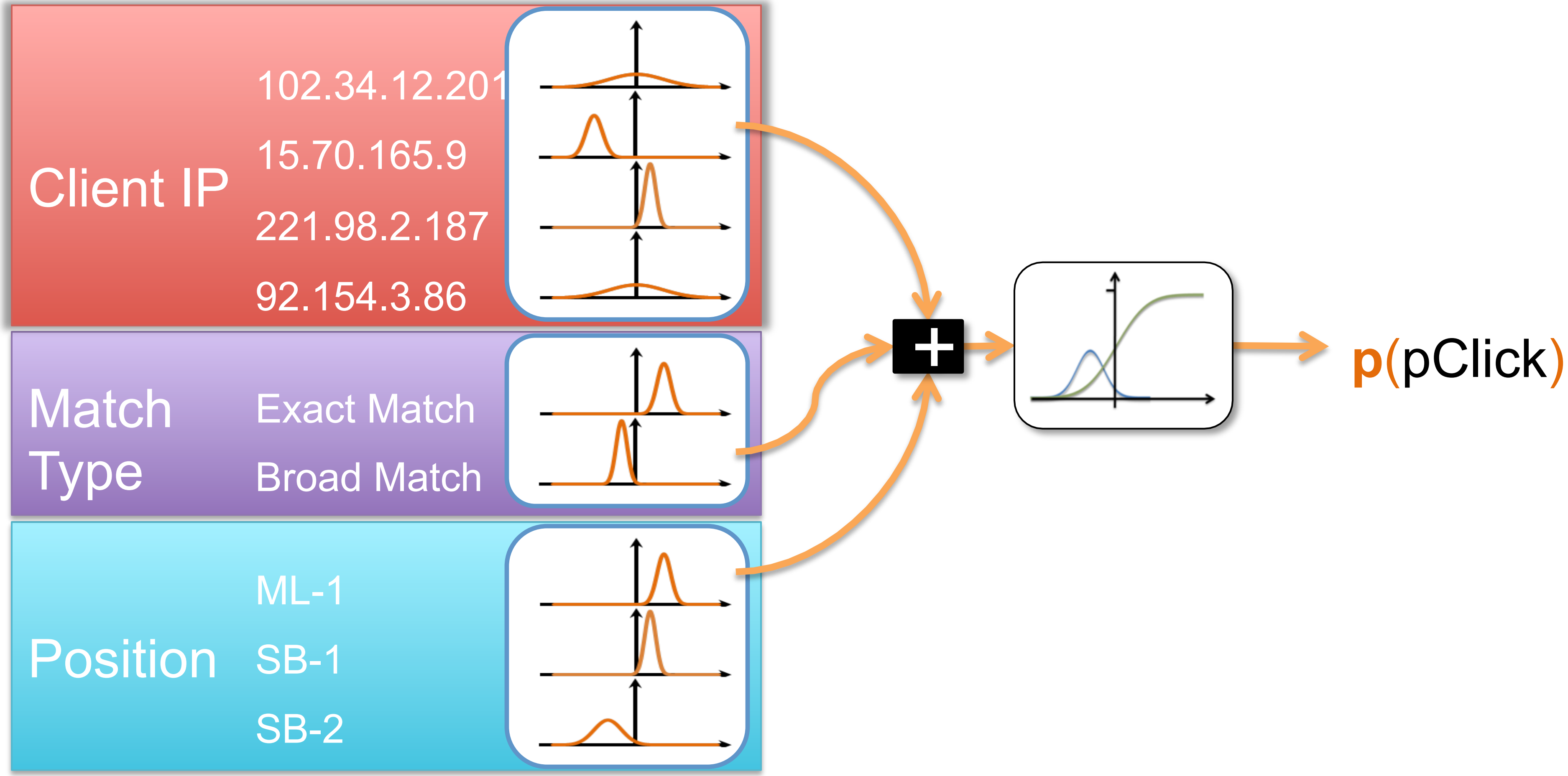
```
/// A single page-view
type PageView =
    {
        ClientDateTime          : DateTime
```

```
/// Different types of media
type MediumType =
    | PaidSearch
    | ContextualSearch
```

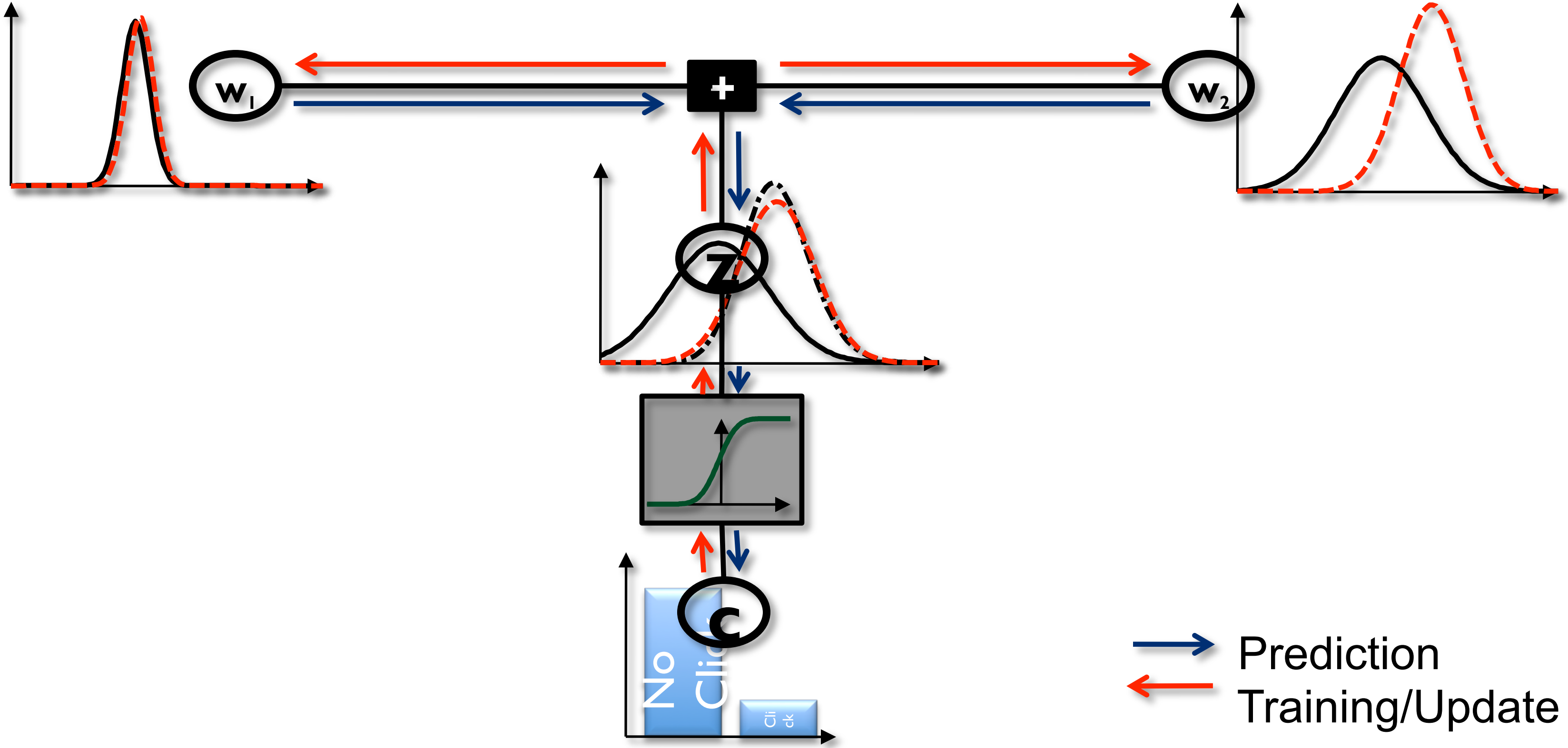
```
/// Create the SQL schema
let schema = bulkBuild ("cpidssdm18", "Cambridge", "June10")

/// Try to open the CSV file and read it pageview by pageview
File.OpenTextReader "HourlyRelevanceFeed.csv"
|> Seq.map (fun s -> s.Split [|','|])
|> Seq.chunkBy (fun xs -> xs.[0])
|> Seq.iteri (fun i (rguid,xss) ->
    /// Write the current in-memory bulk to the Sql database
    if i % 10000 = 0 then
        schema.Flush ()

    /// Get the strongly typed object from the list of CSV file lines
    let pageView = PageView.Parse xss

    /// Insert it
    pageView |> schema.Insert
)
/// One final flush
schema.Flush ()
```

```
        LocationMetroArea       : int
```

Uncertainty: Bayesian Probabilities



Training Algorithm in Action

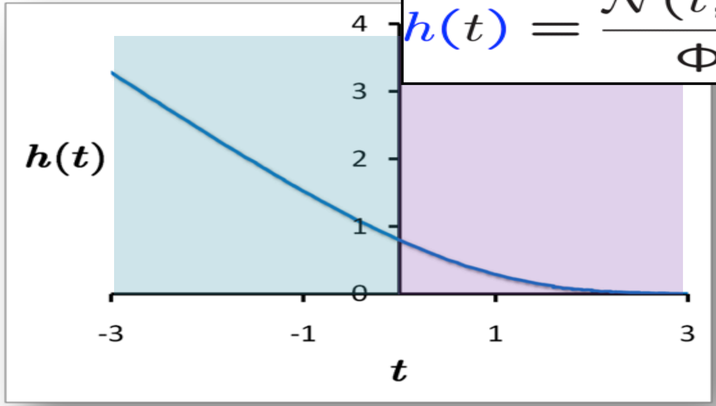


Inference: An Optimization View

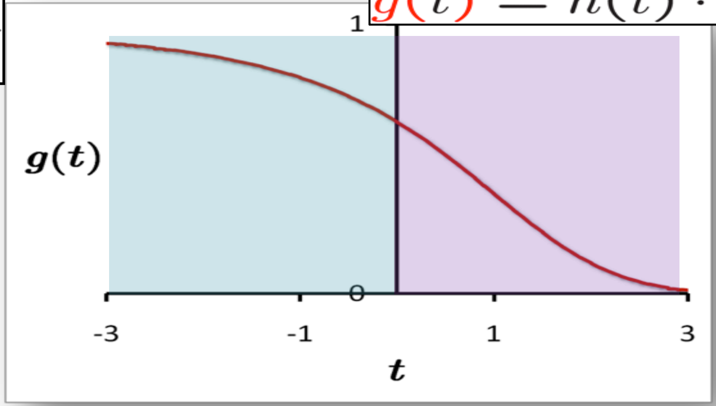
$$\mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{s} \cdot h\left[\frac{\sum_{j=1}^{d}\mu_j}{s}\right] \qquad \sigma_i^2 \leftarrow \sigma_i^2\left(1 - \frac{\sigma_i^2}{s^2} \cdot g\left[\frac{\sum_{j=1}^{d}\mu_j}{s}\right]\right)$$

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

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

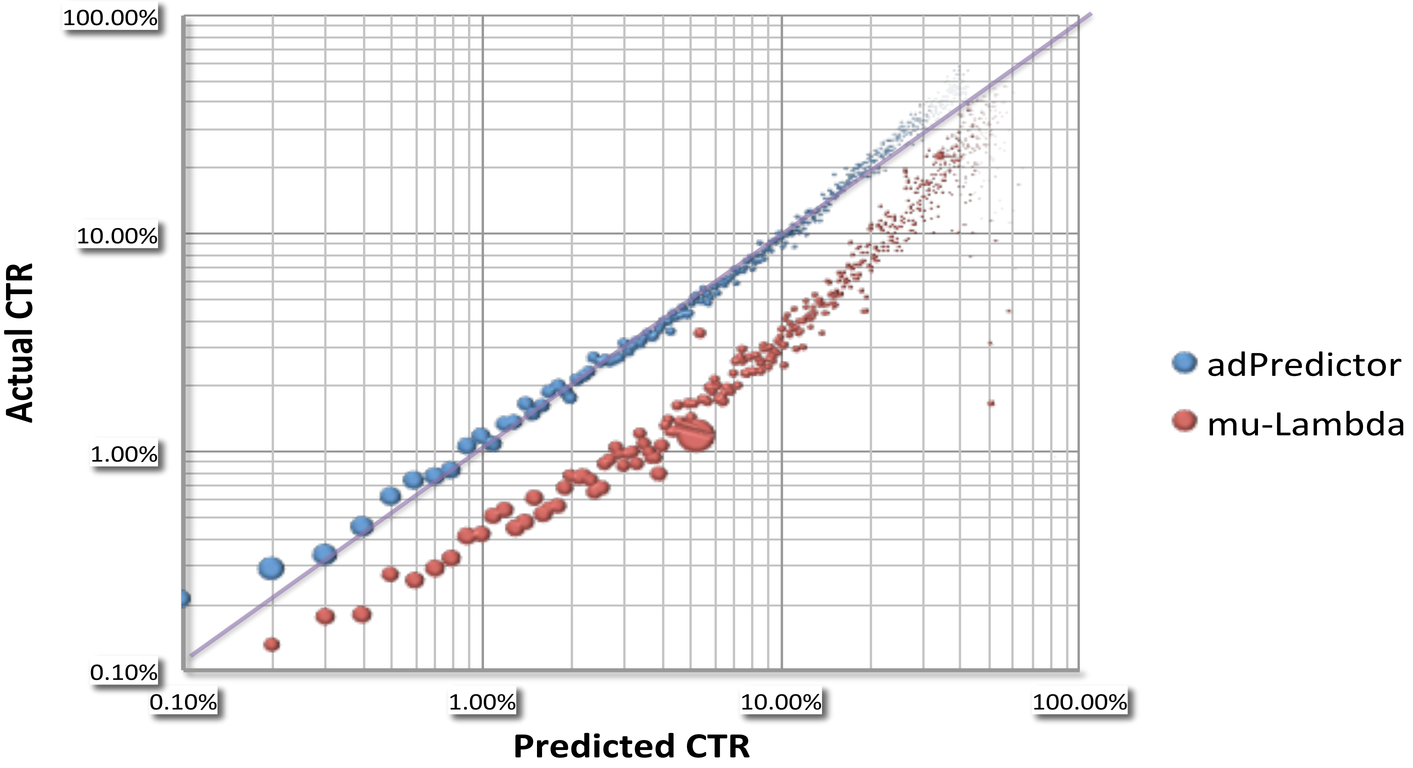


$$g(t) = h(t) \cdot [h(t) + t]$$

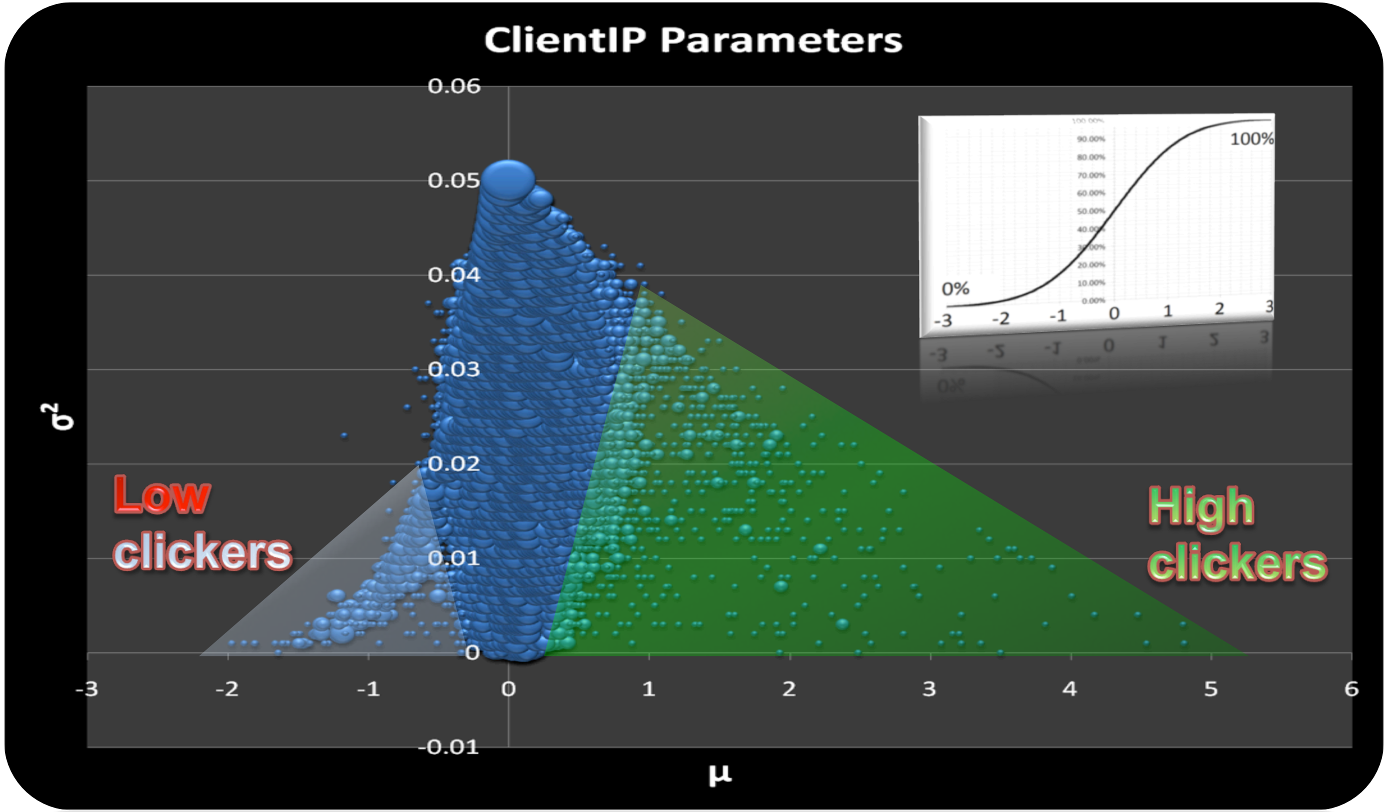


AdPredictor Technology Transfer

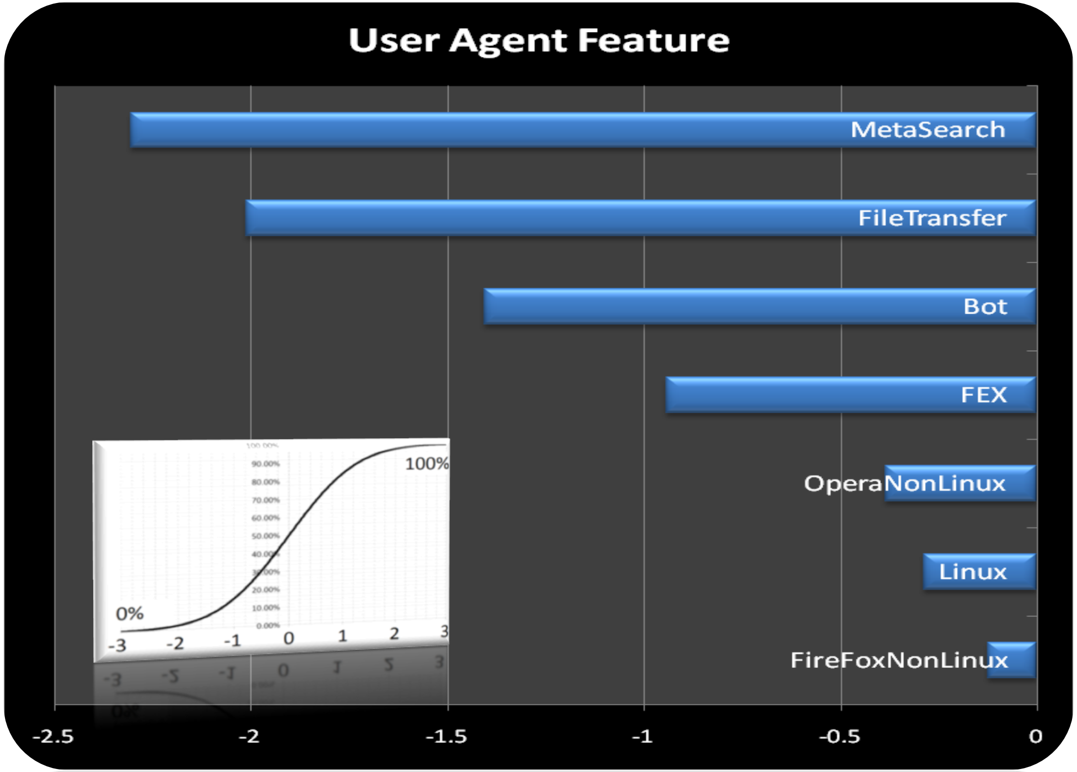
Offline Evaluation



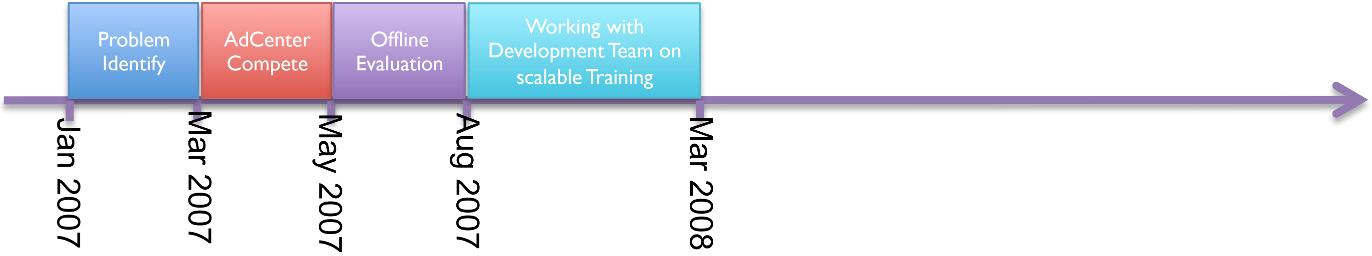
Client IP: Mean & Variance



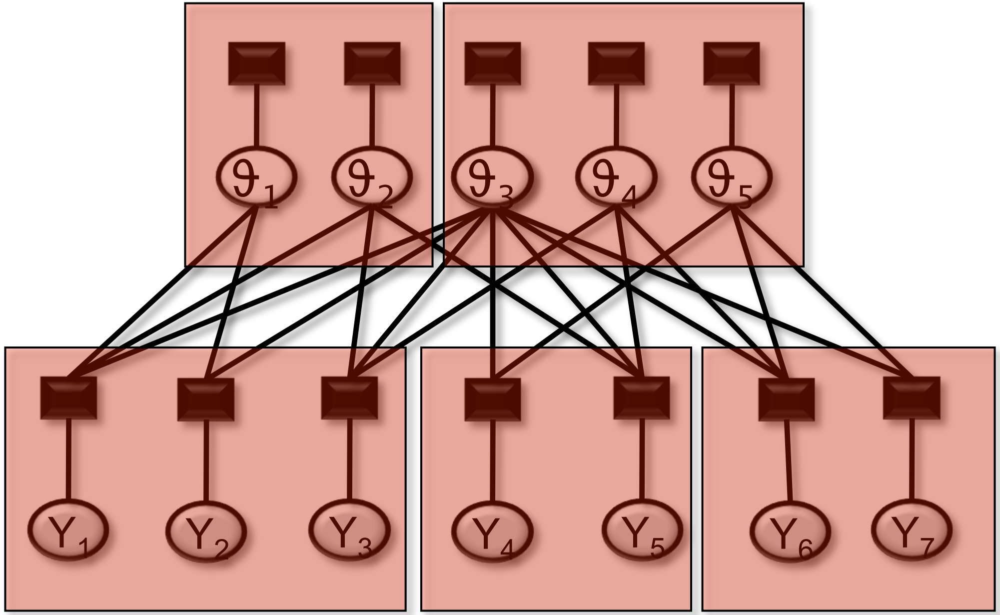
UserAgent: Mean Posterior Effects



AdPredictor Technology Transfer



Distributed Conditional Models



Belief Store (“Memory”)

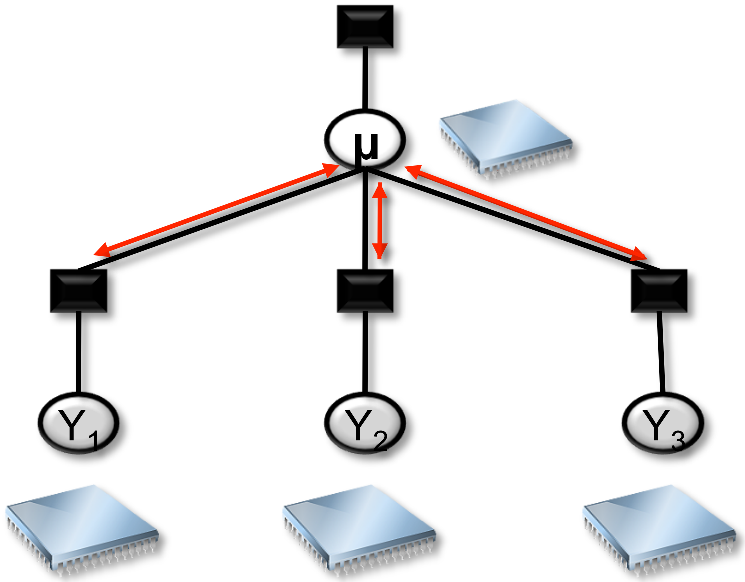
Message Passing (“Communicate”)

Data Messages (“Compute”)

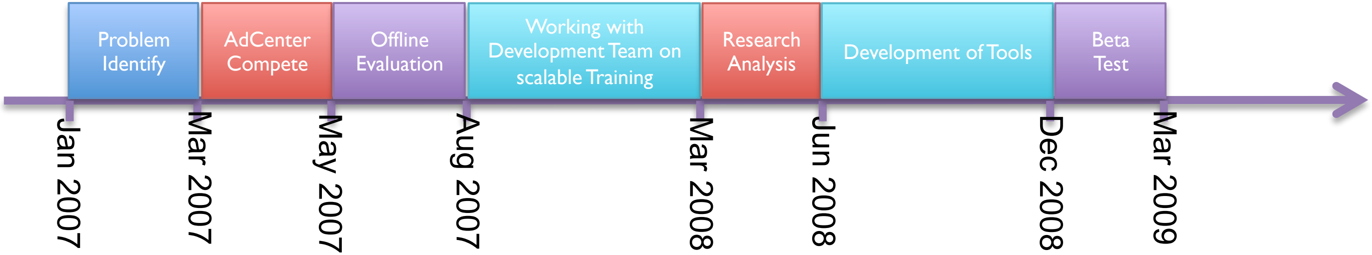
Relation to Map-Reduce

- **Map-Reduce**
  - Map: Data nodes compute messages $m_{F_k \to \theta}$ from data y_i and $m_{\theta \to F_k}$
  - Reduce: Combine messages $m_{F_k \to \mu}$ into $p(\theta)$ by multiplication
  - Vanilla MR is a single pass only!
- **Caveats**:
  - Approximate data factors need all incoming message $m_{F_k \to \theta}$!
  - Each machine needs to be able to store the belief over θ

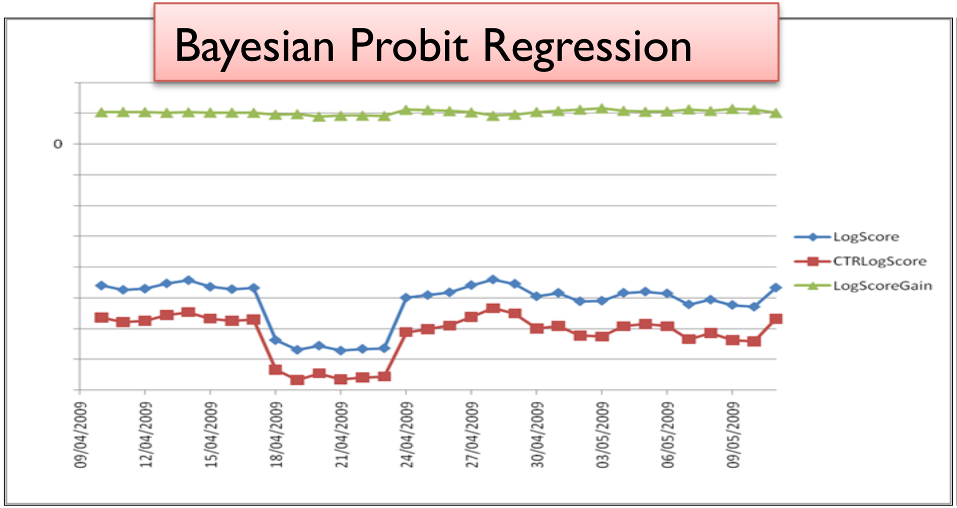
$$p(\boldsymbol{\theta}|\mathbf{x},\mathbf{y}) \propto \prod_k f_k(\mathbf{Y}_k|\boldsymbol{\theta},\mathbf{X}_k) \cdot p(\boldsymbol{\theta})$$

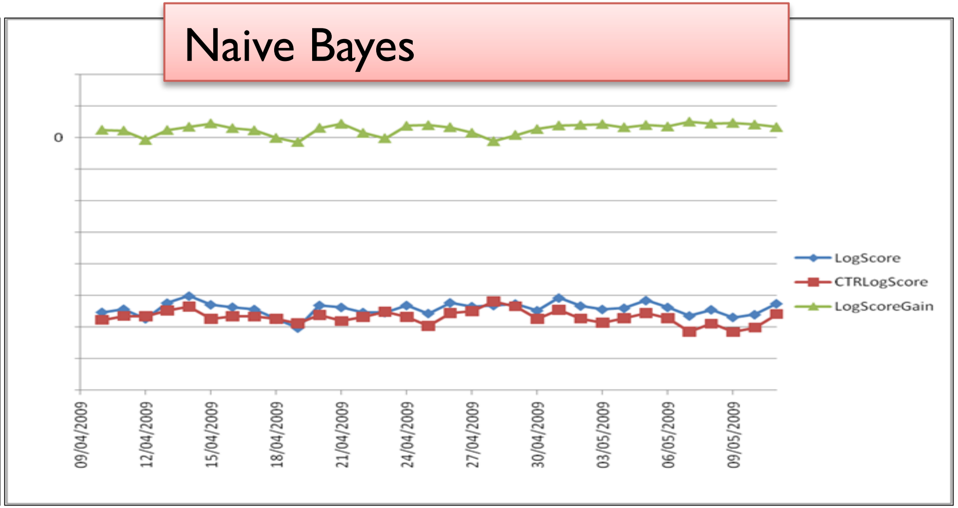


AdPredictor Technology Transfer

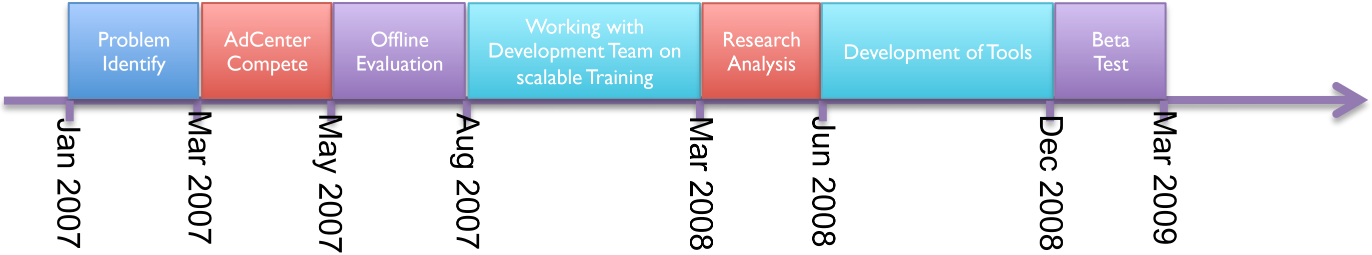


Online Metrics



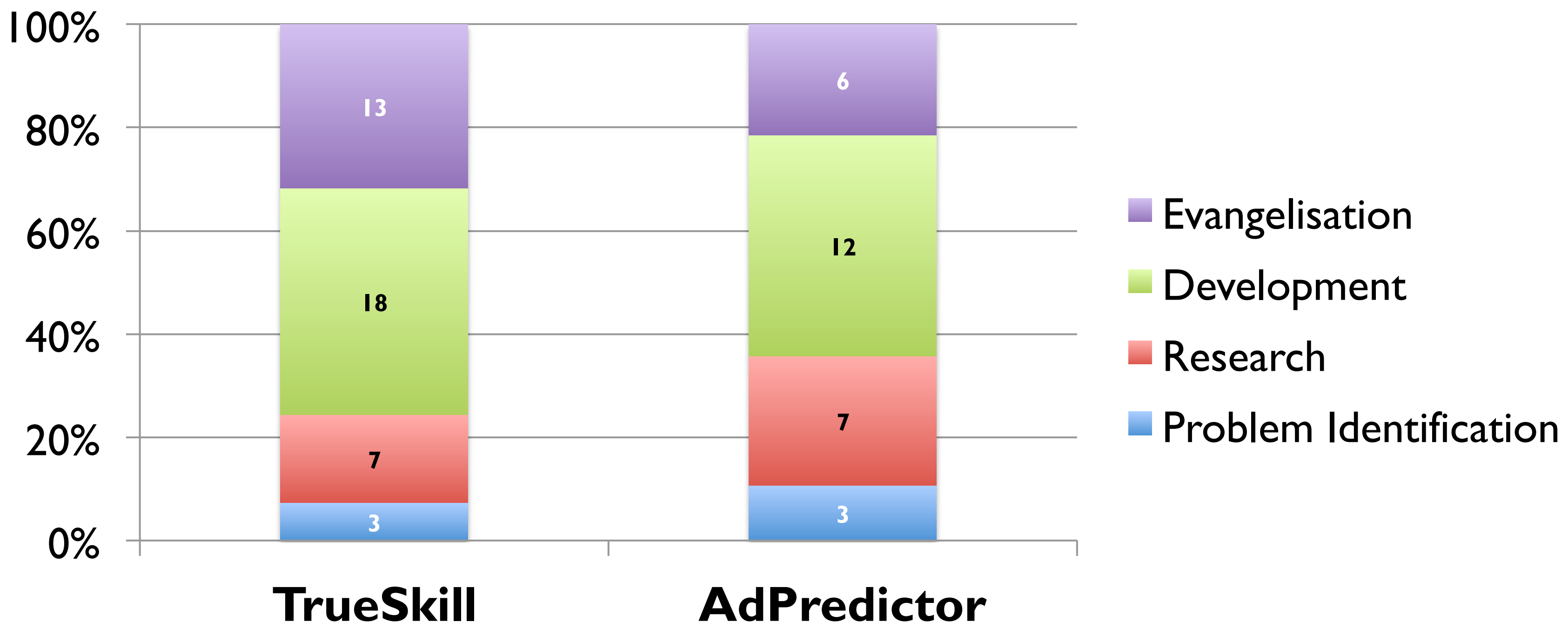


AdPredictor Technology Transfer



- **Lessons Learned:**
  1. Pure research takes a short amount of time
  2. Development takes much longer than planned
  3. Metrics are are important and part of the transfer
  4. Develop for scale from Day 1

Technology Transfer in Numbers



Overview

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

Process Lessons

Problem vs. Technique

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

Try Out and Then Study

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

Under-Promise and Over-Deliver

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

Coding

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

Simplicity

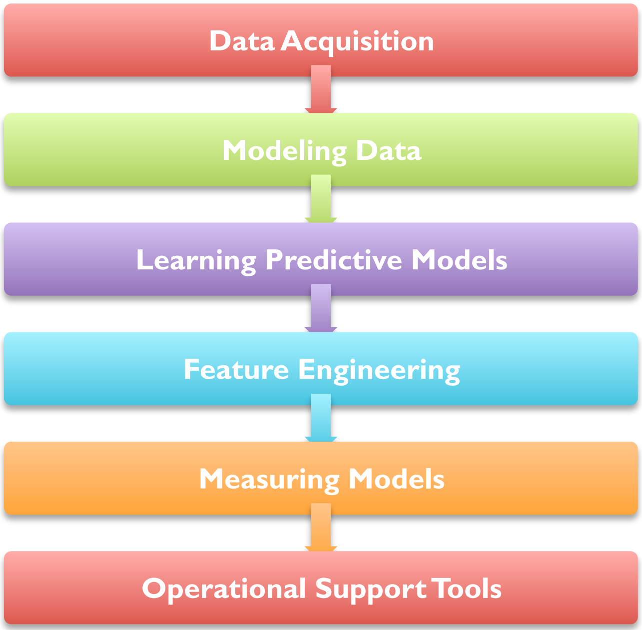
- Simplify to its bare minimum.
- Develop tools and help adoption with customers

Long Run

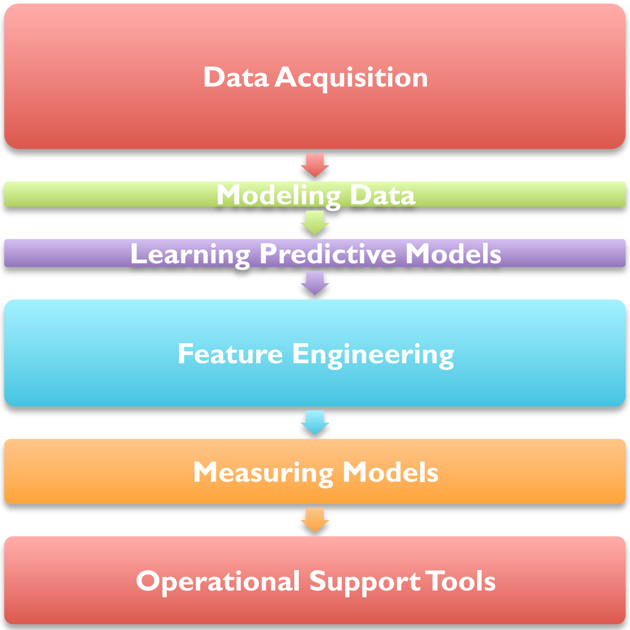
- Be in for the long haul – years is normal
- Do not aim for a quick win!

Process Lessons: Pictures

Steps



Time Allocation



Technical Lessons: Practical Problems

Ideal

- Business case well defined
- Data pipeline established
- Training set given
- Business metric/loss given
- Meaning of data fields fixed
- Breakthrough impact through ML algorithm

Reality

- Business case unclear
- Irregular data-file drops
- No training set
- Unclear measure impact
- Missing & inconsistent data
- ML algorithm leads to single digit improvement at best

How to Pick a Practical ML Problem

- **Key Questions:**
  1. **Data**: Will we have sufficient and ongoing data?
  2. **Complexity**: Can a simple rule do as well?
  3. **Customer Experience**: How will the customer see the prediction/summarization? What will it impact?
  4. **Economics**: What's the cost and benefit of a single prediction?

Example of Practical ML Problems

Good

- Click-Through-Rate Prediction
- Demand Forecasting
- Named Entity Extraction
- Fraud Prediction

Bad

- **Data**: Prediction of mushroom/flower types
- **Complexity**: Learning to predict imputed data
- **Customer Experience**: Predictive menus
- **Economics**: Complex, non-linear models for advertising

Conclusion

- 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)
- Machine Learning = "Statistic of Big Data"?