Traditional machine learning

Logistic regression
Neural networks
K-means, mixture of Gaussians
PCA, kernel PCA, ICA, FA
Support vector machines
Deep belief networks
Decision trees and random forests
... many others ...

Model-based machine learning

Goal: a single modelling framework which supports a wide range of models

Traditional:
“how do I map my problem onto a standard algorithm”?

Model-based:
“what is the model that represents my problem”? 
Realisation of model-based ML

Bayesian framework

Probabilistic graphical models

Efficient deterministic inference
Movie recommender demo
Probabilistic graphical models

Maths (M)  Geometry (G)  Algebra (A)

\[ P(M, G, A) = P(M) \cdot P(G|M) \cdot P(A|M) \]

Graph structure captures domain knowledge
Efficient inference

\[
\sum_{x} \sum_{y} xy = x_1 y_1 + x_2 y_1 + x_1 y_2 + x_2 y_2 \\
= (x_1 + x_2)(y_1 + y_2)
\]
Local message-passing

Maths (M)

Geometry (G)  Algebra (A)
What if distributions are intractable?

- True distribution
- Monte Carlo
- Variational Message Passing
  - Loopy belief propagation
  - Expectation propagation
Algorithms → Models

\[ \overline{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \]

\[ S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \overline{x})(x_n - \overline{x})^T \]

\[ Su_i = \lambda_i u_i \]

M. E. Tipping and C. M. Bishop (1997)

C. M. Bishop (1999)
Allergic Sensitisation Model
Comparison with traditional ML

Separation of model and training algorithm
  Auto-generated inference algorithm
Easy extension to more complex situations
  Modify model, use the same inference algorithms
  Flexible as requirements change
Compact code
  Easy to write and maintain
  Transparent functionality
Many traditional methods are special cases
  One simple framework for newcomers to the field
“Big data”

Computational size vs. statistical size
Noisy ranking

Conventional approach to ranking: “Elo”
- single strength value for each player
- cannot handle teams, or more than 2 players
Bayesian Ranking: *TrueSkill™*

R. Herbrich, T. Minka, and T. Graepel; *NIPS* (2006)
Multi-player multi-team model
Sept. 2005;
10s of millions of users;
millions of matches per day
Convergence

![Graph showing convergence of two players: char (Elo) and SQLWildman (Elo) and their TrueSkill™ equivalents. The x-axis represents the number of games, and the y-axis represents the level. The graph compares the performance of the players over time.](image)

Legend:
- red solid line: char (TrueSkill™)
- blue solid line: SQLWildman (TrueSkill™)
- red dashed line: char (Elo)
- blue dashed line: SQLWildman (Elo)
1. Specify your machine learning problem as a probabilistic model in a .NET program (typically 10-20 lines of code).

2. Use Infer.NET to compile the model into optimized runtime code.

3. Run the code to make inferences on your data automatically.

research.microsoft.com/infernet
research.microsoft.com/~cmbishop
Thank you!