Machine Learning Markets

Putting your money where your mouth is

Amos Storkey
Joint work with Jono Millin and Krzysztof Geras
Quote

• Bob Williamson:
  – The deployment of modern machine learning tools is more akin to a craft than an engineering discipline (*polite version while asking for funding*)

  – Machine learning is a cottage industry, not an engineering discipline. Nearly every new problem is solved from scratch. There is no reuse of previous solutions. There is no language to describe problems. Research is technique-oriented rather than problem-oriented. This lack of modularity and composability limits the field advancing. Reinvention is rife. (*blunt version after receiving funding*)
Machine Learning

• Problem 1
  – Every year: 100s new machine learning methods
  – Which is best? For what?
Machine Learning

- Problem 1
  - Every year: 100s new machine learning methods
  - Which is best? For what?

MAP Estimation for Graphical Models by Likelihood Maximization
Self-Paced Learning for Latent Variable Models
Efficient algorithms for learning kernels from multiple similarity matrices with general convex loss functions
Beyond Actions: Discriminative Models for Contextual Group Activities
Functional Geometry Alignment and Localization of Brain Areas
Efficient Relational Learning with Hidden Variable Detection
Learning to combine foveal glimpses with a third-order Boltzmann machine
Categories and Functional Units: An Infinite Hierarchical Model for Brain Activations
Identifying Dendritic Processing
Tiled convolutional neural networks
Cross Species Expression Analysis using a Dirichlet Process Mixture Model with Latent Matchings
Evaluation of Rarity of Fingerprints in Forensics
Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces
Adaptive Multi-Task Lasso: with Application to eQTL Detection
Practical Large-Scale Optimization for Max-norm Regularization
Joint Cascade Optimization Using A Product Of Boosted Classifiers
Learning To Count Objects in Images
Optimal Web-Scale Tiering as a Flow Problem
Feature Construction for Inverse Reinforcement Learning
Convex Multiple-Instance Learning by Estimating Likelihood Ratio
Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification
Individualized ROI Optimization via Maximization of Group-wise Consistency of Structural and Functional Profiles
Towards Holistic Scene Understanding: Feedback Enabled Cascaded Classification Models
b-Bit Minwise Hashing for Estimating Three-Way Similarities
Construction of Dependent Dirichlet Processes based on Poisson Processes
Machine Learning

- Problem 1
  - Every year: 100s new machine learning methods
  - Which is best? For what?

- Gaussian sampling by local perturbations on
  - Implicit Manifold Learning and its application to Classification
  - On the Multiresolution Analysis of Scale-Space
  - Resolving ambiguities of common features in machine learning
  - The Neural Costs of Optimal and Near-Optimal Estimation
  - Efficient algorithms for learning kernels from multiple similarity matrices with general convex loss functions

- Level Image Representation for Scene Classification & Semantic Feature Sparsification
  - Bayesian Active Learning Without Constraints
  - Efficient Optimization for Discriminative Latent Class Models

- A Bayesian Approach to Concept Drift
  - Effects of Synaptic Weight Diffusion on Learning in Decision Making Networks
  - A Logarithmic Derivative Policy Gradient Method for Reinforcement Learning

- Multi-Target Lasso: with Application to eQTL Detection
  - A Multidimensional Wisdom of Crowds
  - Relaxed Clipping: A Global Training Method for Robust Regression and Classification

- A Log-Multiplicative Language Model
  - Apply to Learning From Text
  - An Analysis on Negative Curvature Induced by Singularity in Multi-Label Learning

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model

- A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
  - A Log-Multiplicative Language Model
Monolithic Machine Learning

Reuse of Methods ➔ Sometimes … Reuse of Results ➔ Rare
Monolithic Machine Learning

Solution 1

Reuse of Methods → Sometimes … Reuse of Results → Rare
Monolithic Machine Learning

Reuse of Methods ➔ Sometimes … Reuse of Results ➔ Rare
Monolithic Machine Learning

Reuse of Methods → Sometimes … Reuse of Results → Rare
Monolithic Machine Learning

Reuse of Methods → Sometimes … Reuse of Results → Rare

Solution 4
Monolithic Machine Learning

Reuse of Methods → Sometimes … Reuse of Results → Rare

Solution 5
Compare

Also: Shopping Mall
Interaction requires communication: protocol, interface.
Combining Bayesian Beliefs

- Wilmers. The Social Entropy Process: Axiomatising the Aggregation of Probabilistic Beliefs 2010
Interim

• There is a need for approaches that develop communication, protocol, interaction, modularity and combination in machine learning.

-------------------------

• Prediction market for machine learning.
• Market makers and equilibria
• Probabilistic rational agents
• Recreating Machine Learning approaches
• Practical Issues
Prediction Markets

- A prediction market is a market with an agreed currency and where goods are bets on each of the possible mutually exclusive (future) outcomes of an event.
- The good associated with outcome $i$ pays 1 Grubnick if $i$ happens and 0 Grubnicks otherwise.

- The prices in the market can be seen as a predictive probability.
Market Design

• Financially neutral market
• Machine Learning agents
• Can be other agent types:
  – Agents with privileged information
  – Agents with external loss functions
• Consider rational agents with fixed “beliefs”
• Consider single equilibrium market, then multiple epochs.

• Then consider practical issues
Inferential Agents and Market Equations

\[ s_i^* = s_i(W_i, c) = \arg \max_{s_i} \sum P_i(k)U_i(W_i - s_i^Tc + s_i) \]

Sum of stockholding of agent \( i \) in all \( K \) goods:
\[ \sum_{i} s_i(W_i, c) = 0 \]

Sum of costs of mutually exclusive events:
\[ \sum_{k} c_k = 1 \]

The market itself is financially neutral, as such, it cannot acquire or owe goods, and hence is subject to the constraint that the total number of goods sold is equal to the total number of goods bought. The no-arbitrage assumption ensures that the costs of mutually exclusive events will sum to 1.
Market Equilibria

• The market equilibrium is an optimal fixed point of the market system where:
  – Every agent’s purchases provide optimal utility for the equilibrium price
  – The market equations are satisfied
• Market equilibria are theoretical assumptions.
• In artificial markets they can be controlled for:
  – Market maker $\rightarrow$ optimization procedure
  – Informational constraints to ensure particular dynamics
• In less controlled markets it is an issue of mechanism design.
• The market equilibrium is a probabilistic model.
Utilities

Logarithmic Utility Function:

\[ U_i^L(W_i, c, s_i) = \sum_k P_i(k) \log(W_i - s_i^T c + s_{ik}) \]

A logarithmic utility function does not allow for debt, but has decreasing utility gains for increasing wealth. This has an equilibrium cost of:

\[ c_k = \frac{\sum_i W_i P_i(k)}{\sum_i W_i} \]

This renders a weighted mixture.
Utilities

Exponential Utility Function:

$$U^E_i(W_i, c, s_i) = - \sum_k P_i(k) \exp(-W_i + s_i^T c - s_{ik})$$

An exponential utility function is upper bounded and allows for unlimited assets and unlimited debts. The effective disutility of debt is exponentially growing, whereas the benefits of even increasing assets become marginal. It has an equilibrium cost of:

$$c_k = \frac{1}{Z} \exp \left( \sum_i \Phi_i(k) \right) \propto \prod_{i=1}^{N_A} P_i(k)^{\frac{1}{N_A}}$$

where: $$\Phi_i(k) = \frac{1}{N_A} \log(P_i(k))$$

This renders a product model. Notice that decisions regarding a change in wealth $x$ are independent of the current wealth $W$, as:

$$- \exp(W - x) = - \exp(W) \exp(-x)$$

Tom Heskes: Log Opinion Pools
Isoelastic Utilities

\[ U_i^I(W_i, c, s_i) = - \sum_k P_i(k) \frac{(W_i - s_i^T c + s_i k)^{1-\eta} - 1}{1-\eta} \]

Isoelastic utilities: buying functions are in proportion to wealth

Split an agent's wealth between two new agents with same beliefs: no change

Logarithmic utility: isoelastic with \( \eta = 1 \)

No closed form solution.

Satisfies

\[ c_k \propto \sum_i W_i \frac{C_k \left( \frac{P_i(k)}{c_k} \right)^{1/\eta}}{\sum_{k'} C_{k'} \left( \frac{P_i(k')}{c_{k'}} \right)^{1/\eta}} \]

More later…
Isoelastic Agents

• A market of isoelastic agents with wealth $W_i$ result in an equilibrium of the form:

$$e_k = \left[ \sum_i \frac{W_i}{Z_i} P_i(k)^{(1/\eta)} \right]^\eta$$

• But isoelastic: Take a component, split in two: same model.
• This is the form of alpha-mixtures, but with explicit mixture weight.
Wealth updates

• Consider fixed beliefs.
• Agents act in market and reap benefit
• Wealth updates:
  – Online updates produce Bayesian Model Updates for the model components (collaborative components for isoelastic agents).
  – Batch updates create Mixtures of effective collaborative beliefs
Multiple tests

Multiple test on UCI data with different systems:

- Agents trained on biased data, and subsamples
- Agents trained using different covariates (aka Random Forests)
- Agents trained using different algorithms.

Agent beliefs are conditionals $P(y|x)$

And different utilities:

- Inhomogenous isoelastic
- Logarithmic
- Exponential
Take Home

• In terms of predictive probability:
  – In most cases, all markets are better than the best individual classifier
  – Market aggregation beats random forests (using various probabilistic formulations see e.g. Caruana).
  – Inhomogenous isoelastic settings are better than other standard settings.

• Accuracy equivalent for RF, but significant elsewhere.
Isoelastic Market v Random Forest
Isoelastic v Others
Multivariate settings

• Univariate multinomial case is...
• ...well boring, really.
• I don’t want to work on UCI datasets...
• What can be done in more interesting settings.
Agent only has opinions on the first variable. Belief is augmented by factorial deviation from the consensus. Thus, niche agents follow the consensus, but differ for variables where they have divergent beliefs.

\[ P_i(k) = \frac{1}{Z} F_i(k) c_k \]  
Belief is a factorial deviation from the market price

\[ U_i(W_i, c, s_i) = \sum_k F_i(k) c_k U_i(W_i - s_i^T c + s_{ik}) \] 
Utility function for all niche agents

\[ U_0(W_0, c, s_0) = \sum_k P_0(k) U_0(W_0 - s_0^T c + s_{0k}) \] 
Utility function for a single agent with direct opinions

\[ c_k \propto P_0(k) \prod_{i=1}^{N_A} F_i(k) \] 
Equilibrium price for niche agents with exponential utility functions
Agent only has opinions on the first variable, and so purchases risk free bets in the variables it has no opinions about. Thus, marginal agents bet only where they have strong beliefs.

The utility maximising position for marginal agents becomes:

$$s_i^*(y_i) = \arg \max_{s_i(y_i)} \sum_{y_i} P_i(y_i) U_i \left( W_i - \sum_{y'_i} s_i(y'_i)^T c(y'_i) + s_{ik}(y_i) \right)$$

This does not lead to an easily computable closed form solution for the costs.
Message Passing

An exponential number of goods is practically infeasible. Agents will have opinions on a reduced set. Thus, the market now has only a single good for each variable. We can compute the price for a bet on variable \( k \), so long as we have computed the messages \( A_{ik} \) for all the agents. The new cost then gets passed to all the agents so they can update their buying functions, resulting in new messages \( A_{ik} \).

A niche agent’s belief is now defined as:

\[
P_i(y) = \frac{1}{\mathcal{Z}} F_i(yS_i) c(y) \quad \text{where:} \quad c(y) = \prod_k c_i(y_k)
\]

Messages are:

\[
A_{ik}(y_k) = \sum_{y^k} P_i\left(y^k | y_k\right) \exp \left( (s_i^k)^T \left( c^k / y^k \right) \right)
\]

And the costs can be calculated as:

\[
\frac{c_k}{1 - c_k} = \frac{\prod_i A_{ik}(1)^{\frac{1}{N}}}{\prod_i A_{ik}(0)^{\frac{1}{N}}}
\]
But

- A market is a mechanism for producing a prediction probability.
- An agent is a mechanism for producing a prediction probability.
- A market is an agent in its own right in a larger market setting:
  - Internal Markets, External Markets are forms of hierarchical model.
- If I care about an outcome, I make a market for a good and hedge on that market. Prize Setting
  - Financial disutility of outcomes: sell prediction goods on the outcome to cover risk
Practical Example

Space of prediction goods
(outcomes, any derivatives of outcomes
(e.g. models), whatever)

You are a special player in the market with an
external interest and internal market action

You have a joint action in both domains.
Future

• Mechanism by which independent machine learning methods can interact on a problem
• Decision theoretic. Don’t have to be Bayesian.

• Actually build a market using a market maker
• Use it on real problems!
• Generalisations & Multivariate implementations
• Commodities and derived goods: features and latent variables.
• Firms: internal markets: hierarchical models
• Mechanism design


