Exact Bayesian Pairwise Preference Learning and Inference on the Uniform Convex Polytope

Scott Sanner

Ehsan Abbasnejad

NICTA

ANU

THE AUSTRALIAN NATIONAL UNIVERSITY
Part I: Bayesian Pairwise Preference Learning (BPPL)

Why?
Learning from Preferences

- RecSys: beautiful reason to be Bayesian
  - I don’t know your utility function $U(x)$ over items $x$
  - But your preference feedback allows me to update beliefs in $U(x)$
Bayesian Decision Theory

• Need to recommend item 1 or item 2:

  – If low: make a recommendation
  – If high: elicit more information

• Can analyze expected loss of a decision

Guo and Sanner, AISTATS-09
Bonilla et al, NIPS-10
Preference Feedback Types

• Like / dislike
  – Difficult to learn \( U(x) \) if no dislikes (Facebook)
  – Little comparative data between items

• Likert scales
  – Discrete ratings don’t allow for arbitrary distinctions
  – You rated *Rambo* and *Finding Nemo* the same
    • *Which did you really like better?*

→ Usually in RecSys, we care about *best* item
→ Total rank ordering comes indirectly
Preference Learning Feedback

- Explicit Preference Feedback
  - Choice queries
    - pick favorite item from set
  - Pairwise is special case of 2-item set
    - low cognitive load if items are comparable
  - Allows for arbitrarily fine distinctions
    - reason vNM builds utility from preferences

Viappiani & Boutilier, NIPS-10
Part II: Bayesian Pairwise Preference Learning (BPPL)

Mathematical Setup
Bayesian Utility Model

• Simple multiattribute utility theory (MAUT)
  – D-dimensional item feature vector $\vec{x}$
  – Additively independent feature weighting vector $\vec{w}$

$$u(\vec{x}|\vec{w}) = \sum_{d=1}^{D} \vec{w}_d \vec{x}_d$$

• What are our beliefs in $\vec{w}$?
Response Likelihood Model

- $a \succ b$: the user prefers $a$ to $b$,
- $a \prec b$: the user prefers $a$ to $b$,
- $a \sim b$: the user is indifferent between $a$ and $b$.

\[
P(Q_{ab} = a \succ b | \vec{w}) = \mathbb{I}[u(a|\vec{w}) - u(b|\vec{w}) > \epsilon]
\]
\[
P(Q_{ab} = a \prec b | \vec{w}) = \mathbb{I}[u(b|\vec{w}) - u(a|\vec{w}) > \epsilon]
\]
\[
P(Q_{ab} = a \sim b | \vec{w}) = \mathbb{I}[|u(a|\vec{w}) - u(b|\vec{w})| \leq \epsilon]
\]
Geometric View

• Each preference constrains weights

• What prior? Gaussian? Uniform?
Uniform Prior

• Prior is hyper-rectangular uniform

\[ P(\bar{w}) = \prod_{d=1}^{D} p(w_d) = \prod_{d=1}^{D} U(w_d; -C, C) \]

• Bayesian update is then

\[ P(\bar{w}|R^{n+1}) \propto P(q_{ab}|\bar{w}, R^n)P(\bar{w}|R^n) \]
\[ \propto P(q_{ab}|\bar{w})P(\bar{w}|R^n) \]

Product of linear constraints
Inference Tasks

• Maximum expected utility item:

\[
\arg\max_x \mathbb{E}_w \left[ u(x|w) \right| R^n ] = \arg\max_x \int_w P(w|R^n) \left[ \sum_{d=1}^D w_d x_d \right] dw
\]

• Probability of preference outcomes:

\[
P(q_{ab}|R^n) = \int_w P(q_{ab}|w) P(w|R^n) dw
\]
Part II: Bayesian Pairwise Preference Learning (BPPL)

Exact Closed-form Inference Machinery (contribution)
Polynomial Case Representation

\[ f = \begin{cases} 
\phi_1 & f_1 \\
\vdots & \vdots \\
\phi_k & f_k 
\end{cases} \]

\[ P(x|y, z) = \begin{cases} 
(x < y + 2z) \land (x > y + z) & C \\
(x \geq y + 2z) \lor (x \leq y + z) & 0
\end{cases} \]

Just need constants for BPPL here, but polynomials needed for extensions

Arbitrary combinations of linear constraints
Polynomial Case Operations: $\oplus$, $\otimes$

\[
\begin{align*}
\begin{cases}
\phi_1 : & f_1 \oplus \\
\phi_2 : & f_2 \\
\end{cases}
\end{align*}
\begin{align*}
\begin{cases}
\psi_1 : & g_1 \\
\psi_2 : & g_2 \\
\end{cases}
= ?
\end{align*}
\]
Polynomial Case Operations: $\oplus$, $\otimes$

$$\begin{align*}
\{ \phi_1 : f_1 \oplus \psi_1 : g_1 \} \\
\{ \phi_2 : f_2 \oplus \psi_2 : g_2 \}
\end{align*} = \begin{align*}
\{ \phi_1 \land \psi_1 : f_1 + g_1 \\
\phi_1 \land \psi_2 : f_1 + g_2 \\
\phi_2 \land \psi_1 : f_2 + g_1 \\
\phi_2 \land \psi_2 : f_2 + g_2
\end{align*}$$

- Similarly for $\otimes$
  - Polynomials closed under $+$, $\times$

- What about max?
  - Max of polynomials is not a polynomial 😞
Polynomial Case Operations: max

$$\max \left( \left\{ \phi_1 : f_1, \phi_2 : f_2 \right\}, \left\{ \psi_1 : g_1, \psi_2 : g_2 \right\} \right) = \ ?$$
Polynomial Case Operations: max

\[
\max \left( \left\{ \phi_1 : f_1, \psi_1 : g_1 \right\}, \left\{ \phi_2 : f_2, \psi_2 : g_2 \right\} \right) = \begin{cases}
\phi_1 \land \psi_1 \land f_1 > g_1 : f_1 \\
\phi_1 \land \psi_1 \land f_1 \leq g_1 : g_1 \\
\phi_1 \land \psi_2 \land f_1 > g_2 : f_1 \\
\phi_1 \land \psi_2 \land f_1 \leq g_2 : g_2 \\
\phi_2 \land \psi_1 \land f_2 > g_1 : f_2 \\
\phi_2 \land \psi_1 \land f_2 \leq g_1 : g_1 \\
\phi_2 \land \psi_2 \land f_2 > g_2 : f_2 \\
\phi_2 \land \psi_2 \land f_2 \leq g_2 : g_2
\end{cases}
\]

• Still a piecewise polynomial!
BPPL Inference

• Just the following operations on cases
  – \( \text{case}_1 \oplus \text{case}_2 \)
  – \( \text{case}_1 \otimes \text{case}_2 \)
  – \( \max(\text{case}_1, \text{case}_2) \)
  – \( \min(\text{case}_1, \text{case}_2) \)
  – \( \int_x \text{case}(x) \) Need to define...
Marginalization: ∫

• ∫ conceivably closed in language
  – But how to compute?

\[
\int_{x_1=-\infty}^{\infty} \sum_i \mathbb{I}[\phi_i] \cdot f_i \, dx_1 = \sum_i \int_{x_1=-\infty}^{\infty} \mathbb{I}[\phi_i] \cdot f_i \, dx_1
\]

• Can simply look at individual partition integrals
  • Then \( \sum \Rightarrow \) we can \( \oplus \) results!
Marginalization: \[ \int \]

- First need to determine integration bounds

\[
\int_{x_1=-\infty}^{\infty} \mathbb{1}[\phi_1] \cdot f_1 \, dx_1
\]

\[
\phi_1 := [x_1 > -1] \land [x_1 > x_2 - 1] \land [x_1 \leq x_2] \land [x_1 \leq x_3 + 1] \land [x_2 > 0]
\]

\[
f_1 := x_1^2 - x_1 x_2
\]

What constraints here?
- independent of \(x_1\)
- pairwise UB > LB

UB and LB are symbolic!

How to evaluate?
Definite Integral Evaluation

• How to evaluate integral bounds?

\[
\left[ \int_{x_1} f_1 \right]_{LB}^{UB} = \frac{1}{3} x_1^3 - \frac{1}{2} x_1^2 x_2 \\
\]

\[
LB := \begin{cases} 
  x_2 - 1 > -1 : & x_2 - 1 \\
  x_2 - 1 \leq -1 : & -1 
\end{cases} \quad UB := \begin{cases} 
  x_2 < x_3 + 1 : & x_2 \\
  x_2 \geq x_3 + 1 : & x_3 + 1 
\end{cases}
\]

• Leaf viewed as composition of XADDs!

\[
f_1 \bigg|_{LB}^{UB} = \left[ \frac{1}{3} UB \otimes UB \otimes UB \ominus UB \otimes UB \frac{1}{2} x_2 \right] \\
\otimes \left[ \frac{1}{3} LB \otimes LB \otimes LB \ominus LB \otimes LB \otimes \frac{1}{2} x_2 \right]
\]

Symbolically evaluated!
That’s It!

We now have all of the operations to compute our BPPL updates and queries in exact, closed-form!
Case → XADD

BPPL needs an efficient data structure for
• compact, minimal case representation
• efficient case operations
Maintaining Compact Cases

- **XADDs:**
  - DDs for continuous variables
    (Sanner, UAI-11)

- **Canonical**
  - Remove redundancy, inconsistency
Compactness of (X)ADDS

- Linear in number of decisions
- Case version has exponential number of partitions!
Binary Operations on (X)ADDs

- Why do we order variable tests?
- Enables us to do efficient binary operations...

Result: XADD operations can avoid case enumeration
XADD Maximization

\[ \max( y > 0, x > 0 ) = \]

May introduce new decision tests
Maintaining XADD Orderings

• Max may get variables out of order

Decision ordering (root→leaf)

\[
\max(y > 0, x > 0) = \begin{cases} 
  x > 0 & \text{if } x > y \\
  y > 0 & \text{if } y > 0 \\
  x > 0 & \text{if } x > 0 
\end{cases}
\]

Newly introduced node is out of order!
Correcting XADD Ordering

• Obtain *ordered* XADD from *unordered* XADD
  – key idea: binary operations maintain orderings

\[
\begin{array}{c}
\text{z is out of order} \\
\text{result will have z in order!}
\end{array}
\]

Inductively assume ID\(_1\) and ID\(_0\) are ordered.

All operands ordered, so applying ⊗, ⊕ produces ordered result!
XADD Pruning

Node unreachable – $x + y < 0$ always false if $x > 0$ & $y > 0$

If linear, can detect with feasibility checker of LP solver & prune
Empirical Results

- A few hard (exponential?) inference cases
  - Need to investigate causes further
Future Work – Exact BPPL

• Efficiency and scalability

• More expressive utility / feedback models
  – Noisy responses
  – Generalized additive utility
  – Nonlinear utility
  – Multi-user utility models

• Preference elicitation
  – Comparison to Gaussian models
Thank you!

Questions?