criteo AI Lab

Adrem
Adrem Data Lab
University of Antwerp

Recommender Systems through the lens of decision theory

Flavian Vasile[1], David Rohde[1], Olivier Jeunen[2], Amine Benhalloum[1], Otmane Sakhi[1]

1 Criteo AI Lab
2 University of Antwerp

criteo AI Lab

Adrem
Adrem Data Lab
University of Antwerp

Recommender Systems through the lens of decision theory

Flavian Vasile[1], David Rohde[1], Olivier Jeunen[2], Amine Benhalloum[1], Otmane Sakhi[1]

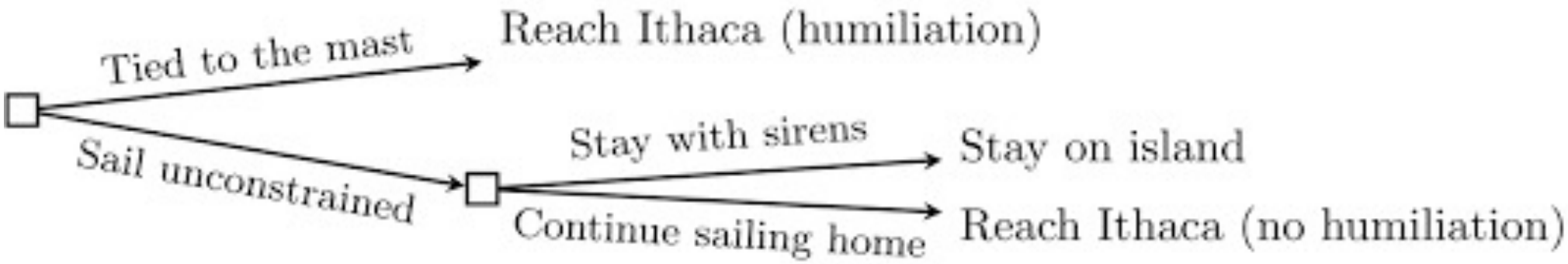
1 Criteo AI Lab
2 University of Antwerp

Thank you to: Alexandre Gilotte, Sergey Ivanov, Mike Gartrell, Ugo Tanielian, Martin Bompaire, Stephen Bonner, the DeepR team and many more…

Part I -Decision Theory Basics

CONTENTS

1. Motivation & Decision Theory (DT) Basics
2. Supervised Learning as DT
3. When actions have consequences – Contextual Bandits (Simple Reco)
4. Confusions in the RecSys literature



Part I. Motivation & Decision Theory Basics

It is often argued that “something special” is needed in a causal setting

.. why?

We will use decision theory to help answer this question.

Decision theory breaks the world into:

- States of nature: describes the possible states of the world
- Decision: maps states to actions
- Utility / Loss: maps the state of nature to a reward
- Models: predict changes of states as a function of actions

It is often argued that “something special” is needed in a causal setting

.. why?

We will use decision theory to help answer this question.

Decision theory breaks the world into:

- States of nature
- Decision
- Utility / Loss
- Models

Empirical risk minimization community focuses on finding the best decision rule.

Optimise to find a good decision rule.

In a causal setting the method breaks down..

It is often argued that “something special” is needed in a causal setting

.. why?

We will use decision theory to help answer this question.

Decision theory breaks the world into:

- States of nature
- Decision
- Utility / Loss
- Models

The Bayesian (or likelihood) community focus on modelling the world.

Optimise to find good model(s).

Nothing changes (at least in principle) in a causal setting.

Decision Theory Basics

What are the **states of nature** that might occur?

Rain / No Rain

Decision Theory Basics

What are the **actions** that we might do?

Carry an umbrella / Leave umbrella

State / Act	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don't carry	Miserable drenching	Bliss unalloyed

State / Act	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don't carry	Miserable drenching	Bliss unalloyed

Leonard "Jimmie" Savage in 1951!

Leonard “Jimmie” Savage in 1951

Act \ State	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don’t carry	Miserable drenching	Bliss unalloyed

Leonard “Jimmie” Savage in 1951

Act \ State	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don’t carry	Miserable drenching	Bliss unalloyed

Leonard “Jimmie” Savage in 1951

Act \ State	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don’t carry	Miserable drenching	Bliss unalloyed

Leonard “Jimmie” Savage in 1951

Act \ State	Rain	Shine
Carry	Inconvenience and wet feet	Inconvenience and slight embarrassment
Don’t carry	Miserable drenching	Bliss unalloyed

How do we decide what to do?

Compute expected utility (or loss) and maximise (or minimise)

THE JOURNAL OF
POLITICAL ECONOMY

Volume LX DECEMBER 1952 Number 6

THE EXPECTED-UTILITY HYPOTHESIS AND
THE MEASURABILITY OF UTILITY

MILTON FRIEDMAN AND L. J. SAVAGE[1]
University of Chicago

RENEWED attention is currently being devoted to a hypothesis about choices involving risk suggested by Gabriel Cramer and Daniel garded as certain, rationalized in terms of consistent preferences for the goods in question and deliberative selection of the alternative highest in the scale of prefer-

Expected Utility

s – State of nature

d – Decision

$U(s, d)$ – Utility function

Expected Utility

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

Expected Utility

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

$$d^* = argmax_d \int U(s, d) P(s) ds$$

Expected Loss

s – State of nature

d – Decision

$L(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

$$d^* = argmin_d \int L(s, d) P(s) ds$$

Umbrella Problem

$$U(s = rain, d = no\ umbrella) = 0$$
$$U(s = rain, d = umbrella) = 1$$
$$U(s = no\ rain, d = no\ umbrella) = 3$$
$$U(s = no\ rain, d = umbrella) = 2$$
$$P(s = rain) = 0.2$$

$$E[U|d = umbrella] =$$
$$U(s = rain, d = umbrella)P(s = rain)$$
$$+ U(s = no\ rain, d = umbrella)P(s = no\ rain)$$

Umbrella Problem

$$U(s = rain, d = no\ umbrella) = 0$$
$$U(s = rain, d = umbrella) = 1$$
$$U(s = no\ rain, d = no\ umbrella) = 3$$
$$U(s = no\ rain, d = umbrella) = 2$$
$$P(s = rain) = 0.2$$

$$E[U|d = umbrella] = 1 \times 0.2 + 2 \times 0.8 = 1.8$$

Umbrella Problem

$$U(s = rain, d = no\ umbrella) = 0$$
$$U(s = rain, d = umbrella) = 1$$
$$U(s = no\ rain, d = no\ umbrella) = 3$$
$$U(s = no\ rain, d = umbrella) = 2$$
$$P(s = rain) = 0.2$$

$$E[U|d = no\ umbrella] =$$
$$U(s = rain, d = no\ umbrella)P(s = rain)$$
$$+ U(s = no\ rain, d = no\ umbrella)P(s = no\ rain)$$

Umbrella Problem

$$U(s = rain, d = no\ umbrella) = 0$$
$$U(s = rain, d = umbrella) = 1$$
$$U(s = no\ rain, d = no\ umbrella) = 3$$
$$U(s = no\ rain, d = umbrella) = 2$$
$$P(s = rain) = 0.2$$

$$E[U|d = no\ umbrella] = 0 \times 0.2 + 3 \times 0.8 = 2.4$$

Frank Ramsey

TRUTH AND PROBABILITY

(1926)

La prévision :
ses lois logiques, ses sources subjectives

par

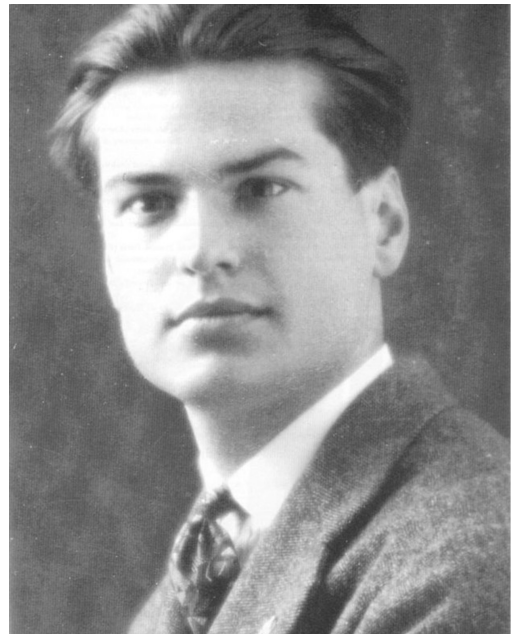
Bruno de FINETTI.

1937.

Frank Ramsey

TRUTH AND PROBABILITY

(1926)



La prévision :
ses lois logiques, ses sources subjectives

par

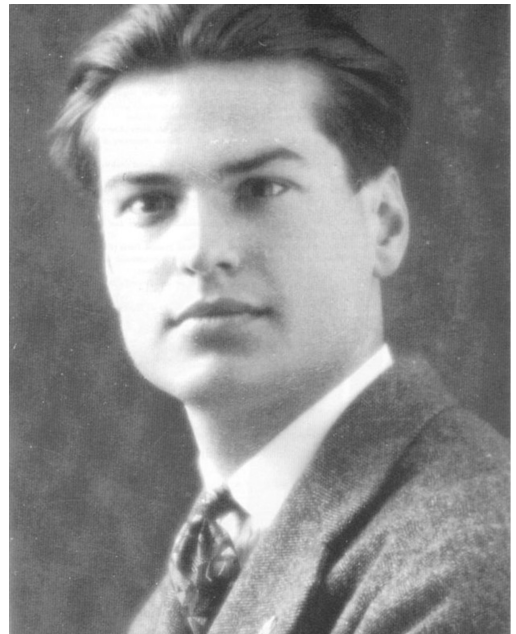
Bruno de FINETTI.

1937.

Frank Ramsey

TRUTH AND PROBABILITY

(1926)



CONTRIBUTIONS TO THE THEORY OF STATISTICAL ESTIMATION AND TESTING HYPOTHESES[1]

By Abraham Wald

Frank Ramsey

TRUTH AND PROBABILITY

(1926)

La prévision :
ses lois logiques, ses sources subjectives

par

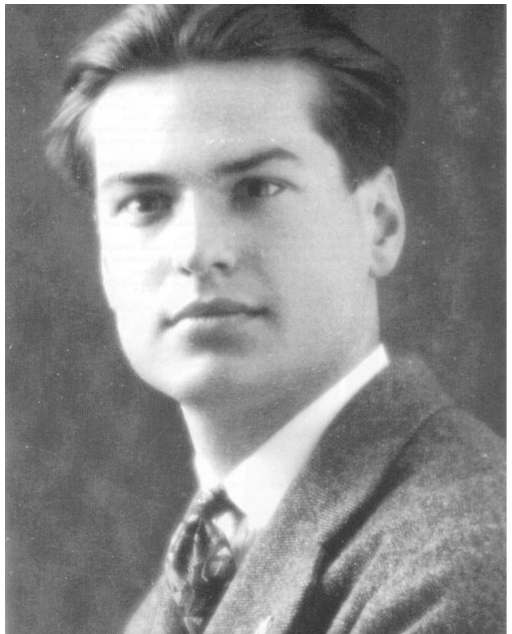
Bruno de FINETTI.

1937.

John von Neumann

Theory of games and economic behavior

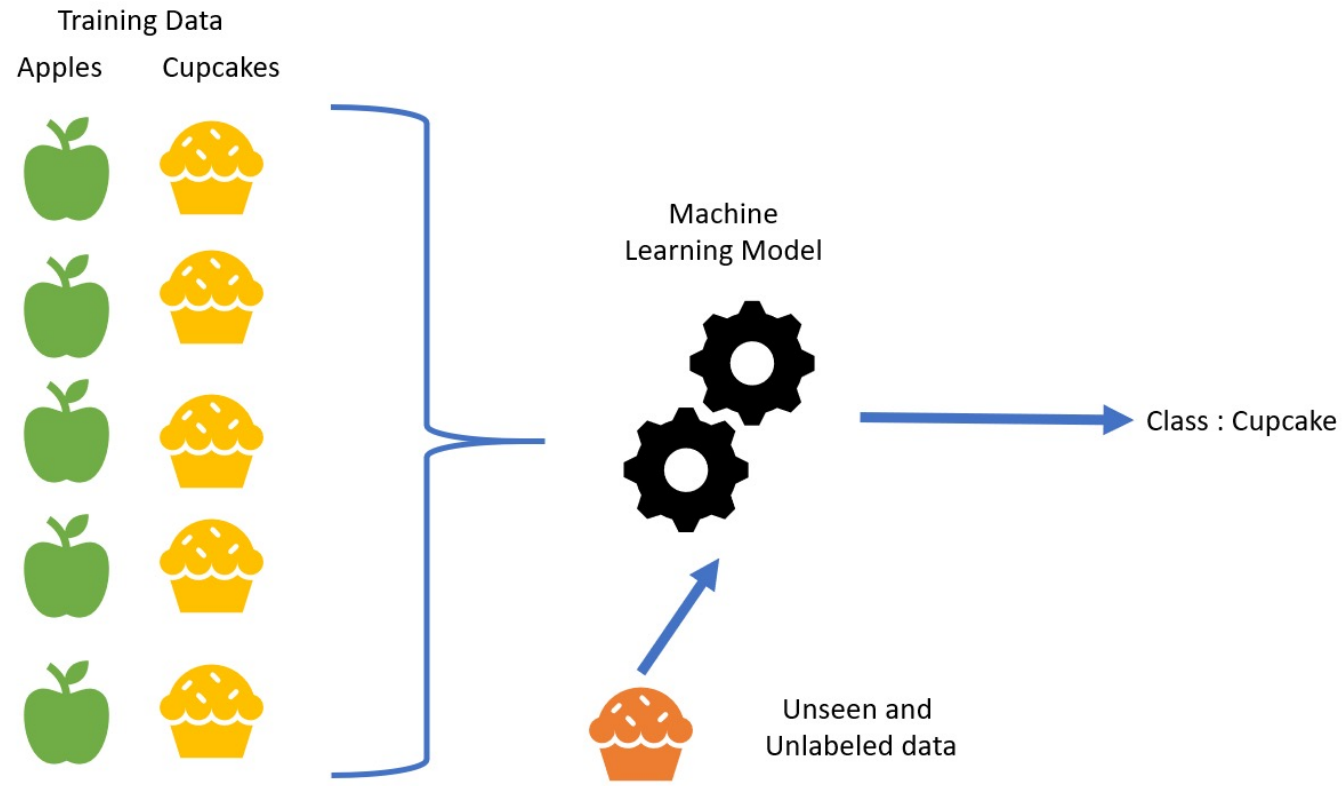
1944



CONTRIBUTIONS TO THE THEORY OF STATISTICAL ESTIMATION AND TESTING HYPOTHESES[1]

By Abraham Wald

Part II. The Easy Case: Supervised Learning



Supervised learning (Focus on the Model)

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

$$d^* = argmax_d \int U(s, d) P(s) ds$$

Approximate the integral using optimisation (maximum likelihood)

Supervised learning (Focus on the Decision)

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

$$d^* = argmax_d \int U(s, d) P(s) ds$$

Solve the argmax using optimisation

Supervised Learning – The Model

$$s = \{y, X\}$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$f(X)$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$f(X)$$

$$L\big(y, f(X)\big) = 1\{y \neq f(X)\}$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta, \Lambda)P(X|\beta, \Lambda)$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta)P(X|\Lambda)$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta)P(X|\Lambda)$$

Likelihood function is then

$$P(D|\beta) = \prod_n P(y_n|X_n, \beta)$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta)P(X|\Lambda)$$

Likelihood function is then

$$\log P(D|\beta) = \sum_n \log P(y_n|X_n, \beta)$$

$$P(y_{n+1}|X_{n+1}, D) = \int P(y_{n+1}|X_{n+1}, \beta)P(\beta|D)d\beta$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta)P(X|\Lambda)$$

Likelihood function is then

$$\log P(D|\beta) = \sum_n \log P(y_n|X_n, \beta)$$

$$P(y_{n+1}|X_{n+1}, D) \approx P\left(y_{n+1}\middle|X_{n+1}, \hat{\beta}\right), \qquad \hat{\beta} = argmax_\beta P(D|\beta)$$

Supervised Learning – The Model

$$s = \{y, X\}$$

$$P(y, X|\beta, \Lambda) = P(y|X, \beta)P(X|\Lambda)$$

Likelihood function is then

$$\log P(D|\beta) = \sum_n \log P(y_n|X_n, \beta)$$

$$P(y_{n+1}|X_{n+1}, D) \approx P\left(y_{n+1}\middle|X_{n+1}, \hat{\beta}\right), \qquad \hat{\beta} = argmax_\beta P(D|\beta)$$

Decision rule $f(X) = 1$ if $P(y_{n+1}|X_{n+1} = X, D) > 0.5$ and otherwise 0

Supervised Learning – The Decision Rule

Search for a good $f_{\Xi}(X)$

Supervised Learning – The Decision Rule

Search for a good $f_{\Xi}(X)$

Unbiased estimator of loss is:

$$\text{Loss} \approx \frac{1}{n}\sum_n L(y_n, f_{\Xi}(X_N))$$

Supervised Learning – The Decision Rule

Search for a good $f_{\Xi}(X)$

Unbiased estimator of loss is:

$$\text{Loss} \approx \frac{1}{n}\sum_n L(y_n, f_{\Xi}(X_N))$$

$$\Xi^* = \text{argmin}_{\Xi} \sum_n L(y_n, f_{\Xi}(X_N))$$

Both approaches

- Model based

$$\beta^* = \text{argmin}_\beta - \sum_n \log P(y_n | X_n, \beta)$$

- Decision rule based

$$\Xi^* = \text{argmin}_\Xi \sum_n L(y_n, \sigma(f_\Xi(X_N)))$$

Both approaches

- Model based

$$\beta^* = \text{argmin}_\beta - \sum_n \text{y}_\text{n} \log \sigma(g_\beta(X_n)) + (1 - y_n) \log(1 - \sigma(g_\beta(X_n)))$$

- Decision rule based

$$\Xi^* = \text{argmin}_\Xi \sum_n L(y_n, \sigma(f_\Xi(X_N)))$$

Both approaches

- Model based

$$\beta^* = \text{argmin}_\beta - \sum_n \text{y}_\text{n} \log \sigma(g_\beta(X_n)) + (1 - y_n) \log(\sigma\left(1 - g_\beta(X_n)\right))$$

- Decision rule based

$$\Xi^* = \text{argmin}_\Xi - \sum_n y_n \sigma(f_\Xi(X_N) + (1 - y_n)(1 - \sigma(f_\Xi(X_n))$$

Both approaches

- Model based

$$\beta^* = \text{argmin}_\beta - \sum_n \text{y}_\text{n} \log \sigma(g_\beta(X_n)) + (1 - y_n) \log(\sigma\left(1 - g_\beta(X_n)\right))$$

- Decision rule based

$$\Xi^* = \text{argmin}_\Xi - \sum_n y_n \log \sigma(f_\Xi(X_N)) + (1 - y_n) \log(1 - \sigma(f_\Xi(X_n)))$$

Both approaches

- Model based

$$\beta^* = \text{argmin}_\beta - \sum_n \text{y}_\text{n} \log \sigma(g_\beta(X_n)) + (1 - y_n) \log(\sigma\left(1 - g_\beta(X_n)\right))$$

- Decision rule based

$$\Xi^* = \text{argmin}_\Xi - \sum_n y_n \log \sigma(f_\Xi(X_N)) + (1 - y_n) \log(1 - \sigma(f_\Xi(X_n)))$$

Cross Entropy Loss is popular, but it might not be the best loss for decision rule optimization, due to the presence of the “log”.

Part III. When actions have consequences…

Decision theory (as per ML textbooks)

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

$$d^* = argmax_d \int U(s, d) P(s) ds$$

Decision theory (as per ML textbooks)

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s)$ – Probability of each state of nature

This only applies to the case where the decision does not change the state of nature.

$$d^* = argmax_d \int U(s, d) P(s) ds$$

Decision theory extension

s – State of nature

d – Decision

$U(s, d)$ – Utility function

$P(s|d)$ – Probability of each state of nature – given decision d

$$d^* = argmax_d \int U(s, d) P(s|d) ds$$

Decision theory extension

s – State of nature

d – Decision

$U(s,d)$ – Utility function

$P(s|d)$ – Probability of each state of nature – given decision d

$$d^* = argmax_d \int U(s,d)P(s|d)ds$$

This extension is in the literature e.g. Phillip Dawid or Joseph Kadane, but it is under discussed

Decision theory extension

s – State of nature

d – Decision

$U(s)$ – Utility function

$P(s|d)$ – Probability of each state of nature – given decision d

$$d^* = argmax_d \int U(s)P(s|d)ds$$

You can simply the form of utility in this case

Simple Reco - Contextual Bandit

$P(s|d) \neq P(s)$ – Causally complex

X – context (user)
a – action (recommendation)
c – reward (click)

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

$\pi(a|\boldsymbol{X})$ - Decision rule

Classification vs Contextual Bandit

Classification

$P(s|d) = P(s)$ – Causally simple

$\boldsymbol{X}$ – features

$\hat{y}$-action (assertion of label)

$L\big(y, f(\boldsymbol{X})\big) = 1\{y \neq f(\boldsymbol{X})\}$ - loss

$f(\boldsymbol{X})$ – Decision rule

$P(y|\boldsymbol{X}, \boldsymbol{\beta})$ - Model

Contextual bandit

$P(s|d) \neq P(s)$ – Causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

Classification

$P(s|d) = P(s)$ – Causally simple

$\boldsymbol{X}$ – features

$\hat{y}$-action (assertion of label)

$L\big(y, f(\boldsymbol{X})\big) = 1\{y \neq f(\boldsymbol{X})\}$ - loss

$f(\boldsymbol{X})$ – Decision rule

$P(y|\boldsymbol{X}, \boldsymbol{\beta})$ - Model

The decision rule is a mapping: $S(X) \rightarrow \{0,1\}$

The model is a mapping: $S(X) \rightarrow [0,1]$

They are not the same but they are similar.

Here $S(X)$ is the support of X.

Classification vs Contextual Bandit

The decision rule is a mapping:
$S(X) \rightarrow S(a)$

The model is a mapping:
$S(X) \times S(a) \rightarrow [0,1]$

They are now completely different!

Here $S(X)$ is the support of X, and $S(a)$.

Contextual bandit

$P(s|d) \neq P(s)$ – Causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

It is often highlighted the fact you only observe the reward for a single action as causing a *paradigm shift*.

i.e. I observe X, a, c but I do not observe X, a', c – maybe a' would have been better or worse..

Contextual bandit

$P(s|d) \neq P(s)$ – causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

This is even called: *The Fundamental Problem of Causal Inference*

Wikipedia:
"The Fundamental Problem of Causal Inference is that it is impossible to observe the causal effect on a single unit. You either take the aspirin now or you don't. As a consequence, assumptions must be made in order to estimate the missing counterfactuals."

Contextual bandit

$P(s|d) \neq P(s)$ – causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

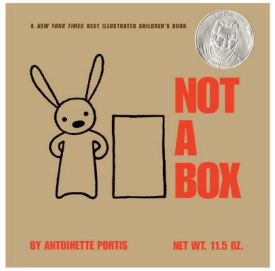
$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

It is often implied this causal setting requires *a paradigm shift..*

Basic Observation #1



This is not a supervised learning problem:

- We don't know the reward of actions not taken—loss function is unknown even at training time.
- Exploration is required to succeed (but still simpler than reinforcement learning – we know which action is responsible for each reward)

Contextual bandit

$P(s|d) \neq P(s)$ – causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

Indeed it isn't possible to apply ERM directly anymore…

Contextual bandit

$P(s|d) \neq P(s)$ – causally complex

X – context (user)

a – action (recommendation)

c – reward (click)

$\pi(a|\boldsymbol{X})$ - Decision rule

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$ - Model

Classification vs Contextual Bandit

Classification **Contextual bandit**

From a modelling point of view almost nothing has changed…. the suggestion that there is a *paradigm shift* or a "*fundamental problem*".. Seems jarring to say the least..

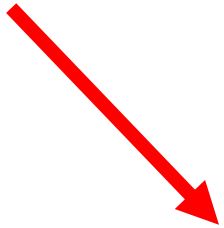
$P(y|\boldsymbol{X}, \boldsymbol{\beta})$ - Model

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$- Model

Classification vs Contextual Bandit

Contextual bandit

In a contextual bandit problem (simplified reco) – *any assertion* that one decision rule is better than another is implicitly an integral over this distribution.



$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$- Model

Classification vs Contextual Bandit

Contextual bandit

- There are 3 main ways to do this:
  A/B Test two or more candidate decision rules under $P(c|\boldsymbol{X}, a)$
- Build a model of $P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$
- Use an estimator of $P(c|\boldsymbol{X}, a)$

$P(c|\boldsymbol{X}, a, \boldsymbol{\beta})$- Model

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n}) \quad \boldsymbol{X} \sim P(\boldsymbol{X})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**

   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$ ← This is so easy, it is hardly mentioned.

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$ ← This is so easy, it is hardly mentioned. We do this for **BLOB**.

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta}) \boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D}) \boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

If we replace $\pi(\cdot)$ with a differentiable model (using a softmax) then we can optimise it directly.

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X}) P(\boldsymbol{X}) d\boldsymbol{X} \approx \frac{1}{N} \sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

We can do this for both $P(c|a, X)$ coming from a model and an IPS based estimator – but it is far more common for IPS based estimators.

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta}) \boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D}) \boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

The use of a softmax makes $\pi(\cdot)$ look like a probability. This might be appealing from the point of view that it motivates the "importance sampling" view of IPS.

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X}) P(\boldsymbol{X}) d\boldsymbol{X} \approx \frac{1}{N} \sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson)**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

CAVEAT: The probabilistic nature of $\pi(\cdot)$ might invite interpretations around:

a) Explore-exploit
b) Hedging-behaviour

.. but neither are justified as is.

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Different approaches for producing a candidate decision rule: $\pi(a|\boldsymbol{X})$

1. **Use a model:**
   Parameterize: $P(c|\boldsymbol{a}, \boldsymbol{X}, \boldsymbol{\beta})$. Then $P(c|a, \boldsymbol{X}, \boldsymbol{D}) = \int P(c|a, \boldsymbol{X}, \boldsymbol{\beta})\boldsymbol{P}(\boldsymbol{\beta}|\boldsymbol{D})\boldsymbol{d\beta}$

   If $\pi(a|\boldsymbol{X})$ is unconstrained then $\pi(\boldsymbol{a}|\boldsymbol{X}) = argmax_a P(c|a, \boldsymbol{X}, \boldsymbol{D})$

2. **Use an unbiased estimator IPS (Horwitz Thompson):**

$$P(c|a, \boldsymbol{X}) \approx \frac{1}{\sum_n 1\{\boldsymbol{X_n} = \boldsymbol{X}\}} \sum_n \frac{c_n 1\{a = a_n, \boldsymbol{X} = \boldsymbol{X_n}\}}{\pi_0(a|\boldsymbol{X})}$$

When applied to the Horwitz Thompson estimator this is known as Counterfactual Risk Minimization (*Swaminathan and Joachims* 2015)

$$E[c|\pi] \approx \int P(c|a = \pi(a|\boldsymbol{X}), \boldsymbol{X})P(\boldsymbol{X})d\boldsymbol{X} \approx \frac{1}{N}\sum_n P(c|a_n = \pi(a_n|\boldsymbol{X}_n), \boldsymbol{X_n})$$

Models or direct methods as described in the literature

“The first – called Direct Modeling (DM) – is based on a reduction to supervised learning, where a regression estimate is trained to predict rewards [2]. To derive a policy, the action with the highest predicted reward is chosen. A drawback of this simple approach is the bias that results from misspecification of the regression model. For real-world data, due to non-linearity or partial observability of the environment, regression models are often substantially misspecified. Hence, the DM approach often does not perform well empirically.”

- "Off-policy Bandits with Deficient Support” *Sachdeva, Su and Joachims(2020)*

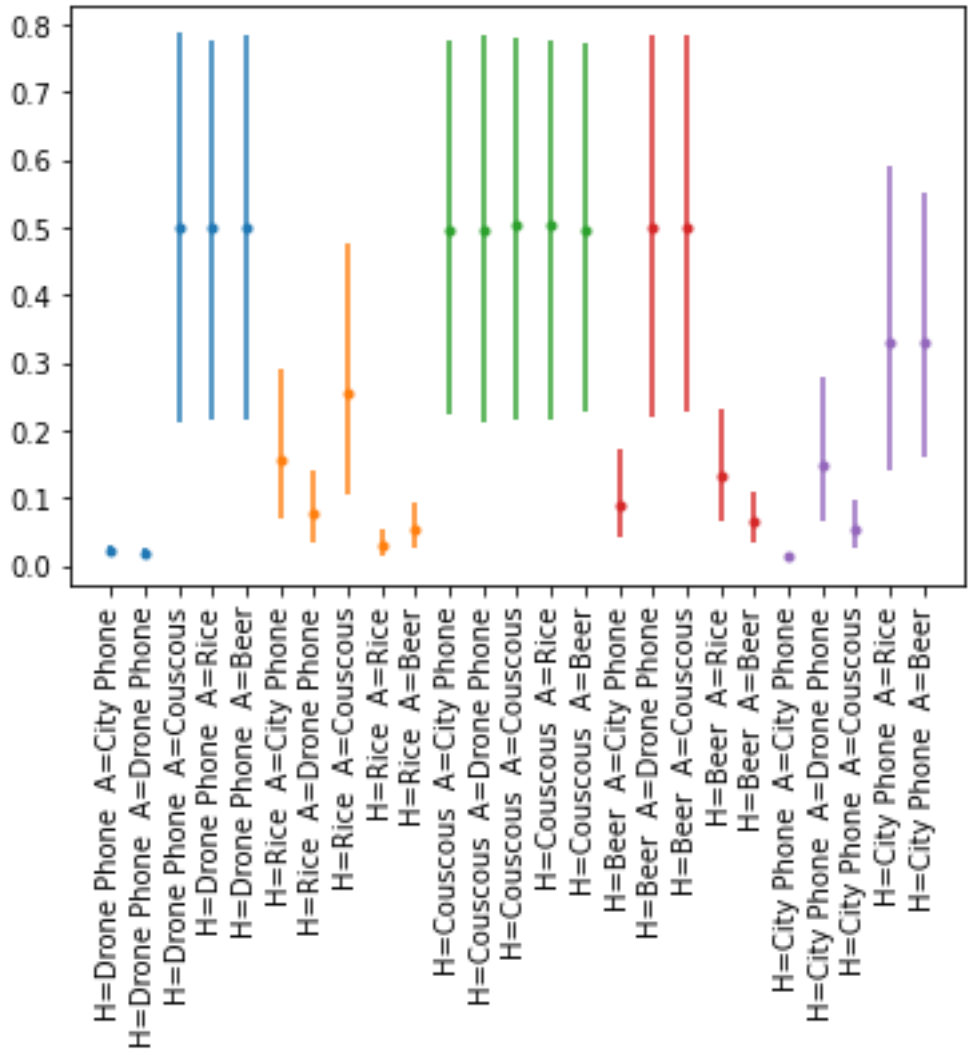
Modelling based approaches can perform very well compared to the IPS estimator

- Large models are not mis-specified
- They do not violate conditionality (substitute expected impressions for impressions)
- They do not rely on an RMS estimator estimator of variance – which can perform badly
- They can “borrow strength” – using action-history, action-action and history-history differences. Parameter sharing can be done to reduce the capacity of the decision rule as per SLT or PAC Bayes.

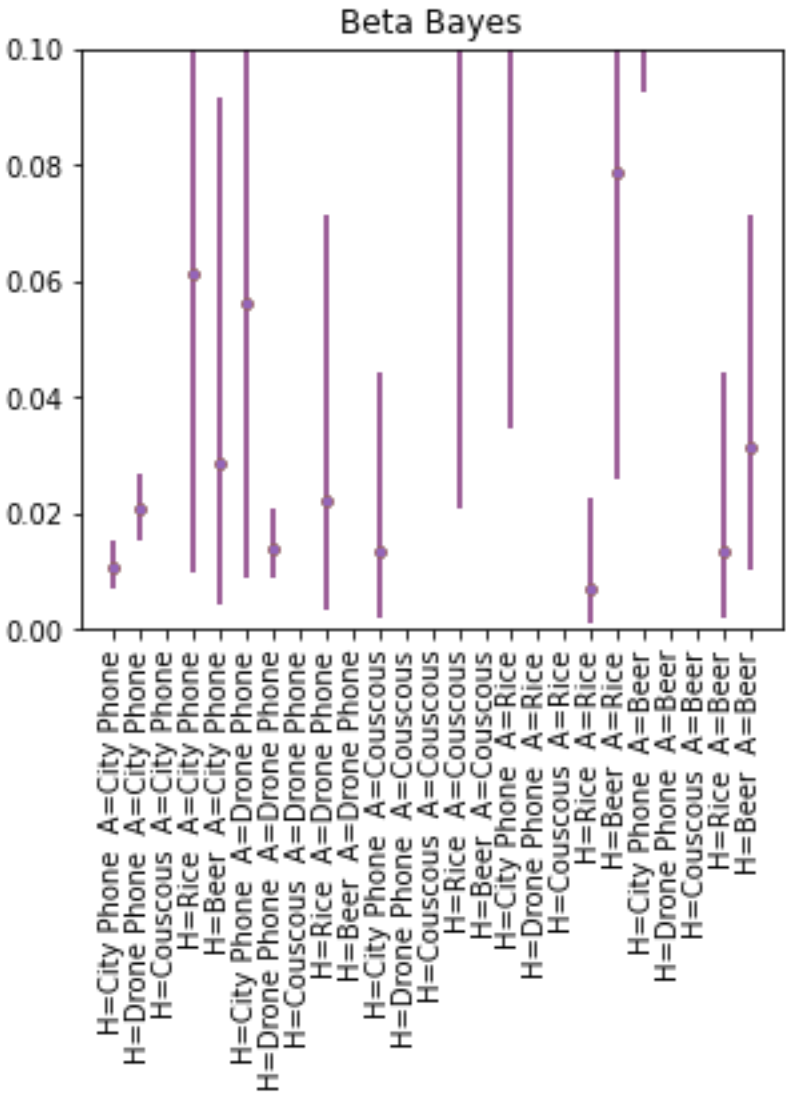
Larry Wasserman – Shows good frequentist performance of Horwitz Thompson compared to Bayes under certain (strong) assumptions (*Robbins Ritrov* Result)

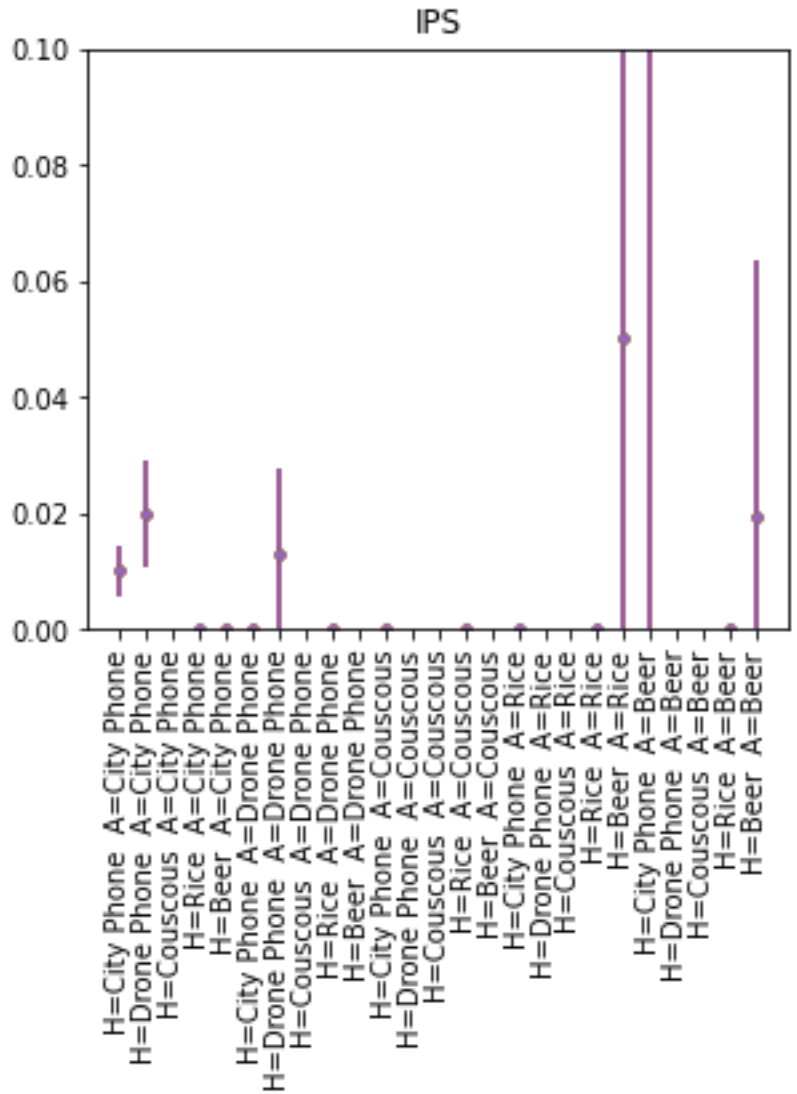
Simple Contextual Bandit Problem

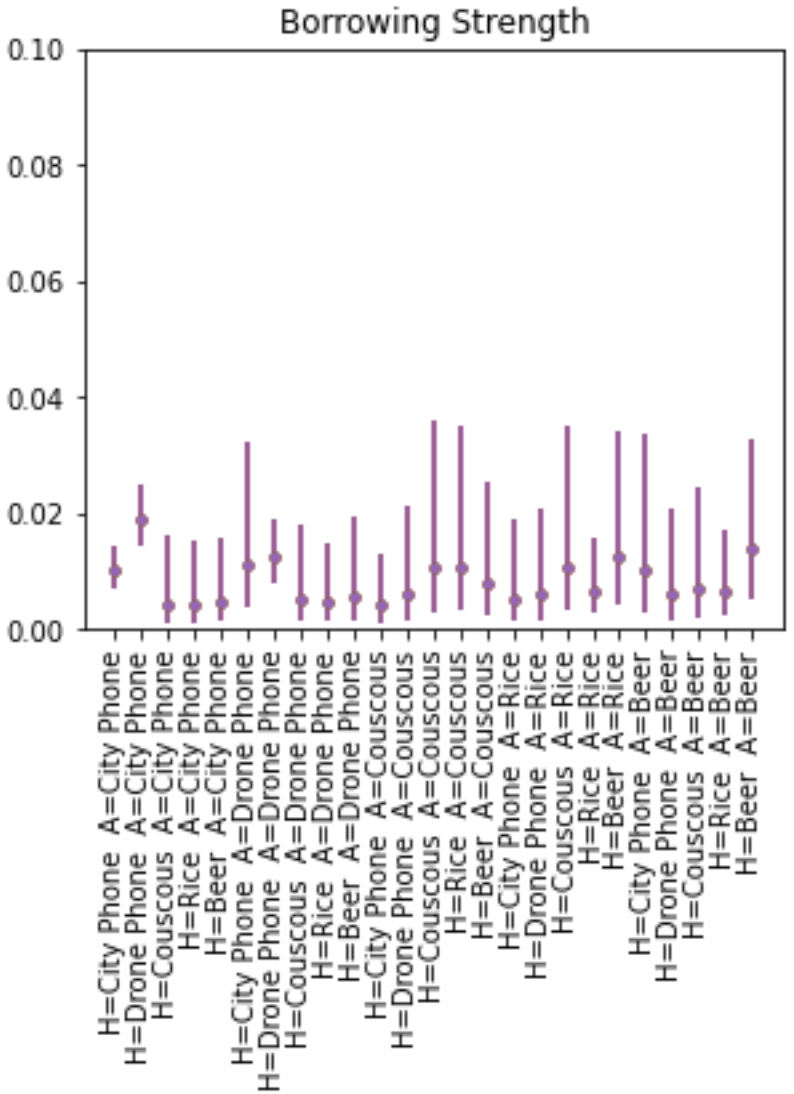
	H	A	C	I
0	City Phone	City Phone	10	1000
1	City Phone	Drone Phone	0	11
2	City Phone	Couscous	0	50
3	City Phone	Rice	0	2
4	City Phone	Beer	1	5
5	Drone Phone	Drone Phone	8	623
6	Drone Phone	City Phone	20	1000
7	Rice	City Phone	0	10
8	Rice	Drone Phone	0	30
9	Rice	Couscous	0	4
10	Rice	Rice	0	100
11	Rice	Beer	0	50
12	Beer	Rice	1	20
13	Beer	Beer	1	52
14	Beer	City Phone	0	23



Simple Contextual Bandit Problem

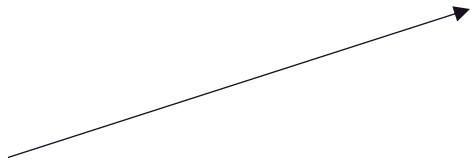






The three fundamental differences of Bandit Reco

$$\boldsymbol{\beta}|\boldsymbol{\Psi} \sim \mathcal{MN}(s^{+}(w_a)\boldsymbol{\Psi}, s^{+}(w_b)\boldsymbol{\Psi}\boldsymbol{\Psi}^{T}, s^{+}(w_b)\frac{1}{P}\boldsymbol{\Psi}^{T}\boldsymbol{\Psi}).$$



Action-History distance

The three fundamental differences of Bandit Reco

$$\boldsymbol{\beta}|\boldsymbol{\Psi} \sim \mathcal{MN}(s^+(w_a)\boldsymbol{\Psi}, s^+(w_b)\boldsymbol{\Psi}\boldsymbol{\Psi}^T, s^+(w_b)\frac{1}{P}\boldsymbol{\Psi}^T\boldsymbol{\Psi}).$$

Action-Action distance

The three fundamental differences of Bandit Reco

$$\beta|\Psi \sim \mathcal{MN}(s^+(w_a)\Psi, s^+(w_b)\Psi\Psi^T, s^+(w_b)\frac{1}{P}\Psi^T\Psi).$$

History-History distance

Part IV.
Confusions in the RecSys literature

Bait and switch in the literature – *It looks like fitting a model, but transitions into optimising a decision rule*

- Softmax

  ERM - Approximation of an argmax when optimising a decision rule

  Model - Transform a vector to the simplex.

- MultiVAE

  Starts off as a modelling approach, but uses heuristics and degenerates to decision rule formulation. Follow-up paper replaces multinomial likelihood with ranking loss.

- Bayesian Personalised Ranking

  Paper: Decision rule is a ranker, imagine the state of nature is a ranking.

- PAC Bayes

  Replace the log likelihood with the negative empirical risk.

Softmax – transform, or approx. argmax

- Modelling approach (transform) two main questions:
  - Compute expectations under $\mathrm{E}_{q(\theta)} \log \sum_n \exp \beta^T \theta_n$
  - Break $\log \sum_n \exp \beta^T \theta_n$ into a sum so that SGD (Robbins Monro) type steps are possible which means sub-sampling the negatives
  - Bouchard bound, Tilted bound, etc.. In RecSys *Bonner and Rohde (2019)*
- Decision rule approach (approx. argmax)
  - Softmax is a differentiable argmax
  - The softmax is slow can we speed it up? Pairwise ranking can be used as a fast surrogate
  - If we want not just the best item, but the best K softmax is no longer correct *Tanielian and Vasile (2018)*

MultiVAE – *Liang et al (2018)*

$$z_u \sim N(0, I)$$
$$v_{u,1}, .., v_{u,n} \sim \text{categorical}(softmax(f_\theta(z_u)))$$

Probabilistic model that assumes a user has a latent representation z_u and items cluster together. If z_u has sufficient dimensionality law of large numbers results.

$$\mathcal{L}_\beta(\mathbf{x}_u; \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)] - \beta \cdot \mathrm{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \| p(\mathbf{z}_u)).$$

(a) ML-20M

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.395**	**0.537**	**0.426**
Mult-DAE	0.387	0.524	0.419
WMF	0.360	0.498	0.386
SLIM	0.370	0.495	0.401
CDAE	0.391	0.523	0.418

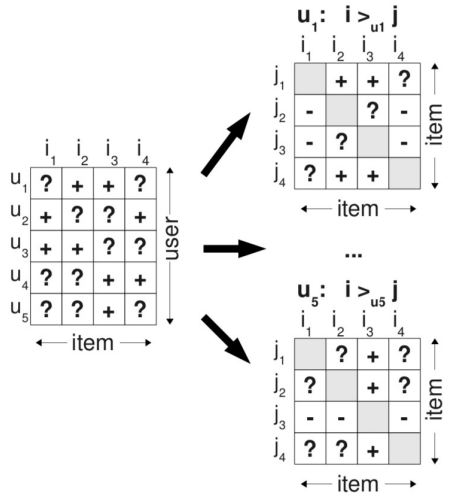
In practice, the variational bound is modified in an ad-hoc way by scaling the KL term and the model is tuned for recall@K ranking.

Bayesian Personalised Ranking (BPR)
Rendle (2009)

- Recommender systems can be viewed as produced personalised ranking for users. This is generally true
- BPR assumes that you observe rankings of items. This is rarely true. Perhaps there is a case with classic RecSys data such as MovieLens.

- In practice BPR is not used in a modelling way, but rather as a ranking loss.

If an item i has been viewed by user u – i.e. $(u, i) \in S$ – then we assume that the user prefers this item over all other non-observed items. E.g. in Figure 2 user u_1 has viewed item i_2 but not item i_1, so we assume that this user prefers item i_2 over i_1: $i_2 >_u i_1$.



PAC Bayes *McAllester (1999)*

PAC Bayes is a (non-Bayesian) extension to statistical learning theory that explicitly replaces the log likelihood with the negative loss.

It is sometimes presented as an extension of Bayes – but from a Bayesian perspective this makes no sense. Bayesian decision theory has clear separation between model and decision rule. PAC Bayes mixes the two concepts.

A PRIMER ON PAC-BAYESIAN LEARNING

by

Benjamin Guedj

(7) $$\pi_\lambda\left(\widehat{\phi}|\mathcal{D}_n\right) \propto \ell_{\lambda,n}\left(\widehat{\phi}\right) \times \pi_0\left(\widehat{\phi}\right),$$

where $\ell_{\lambda,n}$ is a loss term measuring the quality of the predictor $\widehat{\phi}$ on the collected data $\mathcal{D}_n$ (the training data, on which $\widehat{\phi}$ is built upon). To set ideas, one could think of $\ell_{\lambda,n}$ as a functional of the empirical risk r_n.