Models of collective inference

Laurent Massoulié (Microsoft Research-Inria Joint Centre)
Mesrob I. Ohannessian (University of California, San Diego)
Alexandre Proutière (KTH Royal Institute of Technology)
Kuang Xu (Stanford)
Outline

- Spreading news to interested users
  How to jointly achieve news
  - categorization by users, and
  - dissemination to interested users
  [LM, M. Ohanessian, A. Proutière, Sigmetrics 15]

- Information-processing systems
  Treatment of inference tasks by pool of agents with limited capacities (ex: crowdsourcing)
  [LM, K. Xu 15]
A sequential decision problem

- Unknown latent variable $\theta \in T$ (news topic)
- Nodes $u \in U$, if selected, generate feedback $Y_u \sim Ber(p_{u\theta})$ (appreciation of recommended news)
- Goal: sequentially select nodes $U^1, \ldots, U^I$ up to stopping time $I$ to minimize $\mathbb{E}[C_0 + C_1]$

where for $U^I = \{U^1, \ldots, U^I\}$

$$C_0 = \sum_{u \notin U^I} (2p_{u\theta} - 1)_+ \quad \text{(missed opportunities)}$$

$$C_1 = \sum_{u \in U^I} (1 - 2p_{u\theta})_+ \quad \text{(spam)}$$

Assumption: partially known preferences $g_{ut} = 1\{p_{ut} > \frac{1}{2}\}, \ t \in T, \ u \in U$
Related Work

• Multi-Armed Bandits (MAB):
  – Contextual [Li et al. 2010]
  – Infinitely many arms [Berry et al. 1997]
  – Secretary problem [Ferguson 1989]
  – Best arm identification [Kaufmann et al. 2014]

• Distinguishing features:
  – Partial information on parameters
  – Finitely many, exhaustible arms
  – Stopping time, trades off two regrets
  – Find all good arms, not the best
Pseudo-Bayesian Updates

\[ g_{ut} = 1\{p_{ut} > \frac{1}{2}\} \]

- Pseudo-preferences:
  \[ q_{ut} = \frac{1 + r}{2} g_{ut} + \frac{1 - r}{2} (1 - g_{ut}) \]

- Initial prior:
  \[ \nu_t^0 = \frac{1}{|\mathcal{T}|} \]

- Pseudo-posterior update:
  \[
  \nu_t^{i+1} = \frac{\nu_t^i[q_{U_{it}}X_i + (1 - q_{U_{it}})(1 - X_i)]}{\sum_{s \in \mathcal{T}} \nu_s^i[q_{U_{is}}X_i + (1 - q_{U_{is}})(1 - X_i)]}
  \]
Greedy-Bayes Algorithm

- Initialize pseudo-preferences and prior
- At each step $i$
  - Make greedy selection $U^i \in \arg\max_{u \neq U^1, \ldots, U^{i-1}} \sum_{t \in \mathcal{T}} g_{ut} \nu^i_t$
  - Based on response $X^i$, update pseudo-posterior $\nu^{i+1}_t, \forall t \in \mathcal{T}$
  - Stop if $\nu^{i+1}\left\{ t : \exists u \notin U^i, g_{ut} = 1 \right\} \leq \epsilon^{\text{threshold}}$
How good is Greedy-Bayes?

• Separation: \[ |p_{ut} - \frac{1}{2}| \geq \delta \quad \forall u, t \]

• Main Theorem:
  – If \( r = \Theta(\delta) \) and \( \log \frac{1}{\epsilon} = \Omega \left( \log (|U| \cdot |T|) \right) \)
  – Then, under Greedy-Bayes:
    \[
    \mathbb{E}[C_0 + C_1] = O \left( \frac{|T|}{\delta^2} \log (|U| \cdot |T|) \right)
    \]
  – Moreover \( p_{ut} \) can be such that for any algorithm (uniformly at least as good as Greedy-Bayes):
    \[
    \mathbb{E}[C_0 + C_1] = \Omega \left( \frac{|T|}{\delta} \log (|U|) \right)
    \]
The tool: Bernstein’s inequality for martingales

[Freedman’75]

Martingale $\{M^i\}_{i \geq 1}$ with compensator $\{\langle M \rangle^i\}$ and increments bounded by $r$ satisfies for any stopping time $I$

$$\forall L, z > 0, \ P \left( \max_{i \leq I} |M^i| \geq z, \langle M \rangle^I \leq L \right) \leq 2e^{-\frac{z^2}{2(L + rz/3)}}$$
Proof sketch

bound on spamming regret $C_1$

\[ C_1 \leq \sum_{i=1}^I 1\{\text{spam at } i\} \leq |U| \]

Goal:

\[ \mathbb{P} \left\{ \sum_{i=1}^I Z_i > x \right\} = o \left( \frac{1}{|U|} \right) \]

Key quantity:

\[ R^i = \log \frac{\nu^i_{\Theta}}{\nu^i \{ t : \exists u \notin U^i, g_{ut} = 1, g_{u\Theta} = 0 \}} \]

\[ \mathbb{P} \left\{ \sum_{i=1}^I Z_i > x \right\} = \sum_{i=1}^{\left| U \right|} \mathbb{P} \left\{ Z_i = 1, \sum_{j=1}^i Z_j = x \right\} \]

\[ \leq \sum_{i=1}^{\left| U \right|} \mathbb{P} \left\{ R^i \leq \log(1/\epsilon), \sum_{j=1}^i Z_j = x \right\} \]
Proof sketch
bound on spamming regret $C_1$

Martingale analysis of $R^i : R^i = A^i + M^i$ where $M^i$ martingale with increments bounded by $O(\delta)$, and for some $\tilde{Z}^i \geq Z^i$:

$$A^i \geq \Omega \left( \frac{\delta^2}{\tau} \sum_{j=1}^{i} \tilde{Z}^j \right),$$
$$\langle M \rangle^i \leq O \left( \frac{\delta^2}{\tau} \sum_{j=1}^{i} \tilde{Z}^j \right).$$

Hence:

$$P \left\{ R^i \leq \log(1/\epsilon), \sum_{j=1}^{i} Z^j = x \right\} \leq \sum_{\ell \geq x} P \left\{ R^i \leq \log(1/\epsilon), \lfloor \sum_{j=1}^{i} \tilde{Z}^j \rfloor = \ell \right\} \leq \sum_{\ell \geq x} P \left\{ M^i \leq -\Omega(\delta^2 \ell/\tau) + \log(1/\epsilon), \ldots \right\} \leq \sum_{\ell \geq x} P \left\{ \langle M \rangle^i \leq O(\delta^2 \ell/\tau) \right\} \rightarrow \text{then apply Freedman’s inequality}$$
Lower regret with extra structure

• Topic that many like who dislike other topics

\((*)\) \(\forall t \neq \theta, |\{u : g_{u\theta} = 1, g_{ut} = 0\}| \geq c \log |U|\)

• **Theorem**: Under \((*)\), if \(c = \Omega\left(\frac{1}{\delta^2}\right)\) then for topic \(\theta\)

Greedy-Bayes has regret \(O\left(\frac{|T|}{\delta^2}\right) \rightarrow \text{Constant in } |U|\)

• Moreover \(p_{ut}\) can be such that \(E[C_0 + C_1] = \Omega\left(\frac{|T|}{\delta}\right)\)

for any algorithm (uniformly at least as good as Greedy-Bayes)

• Other conditions give \(\log|T|\) instead of \(|T|\)
Greedy-Bayesian Algorithm

• Given $|U| \times |T|$ matrix $g_{ut} = 1\{p_{ut} > \frac{1}{2}\}$:
  – Initialize pseudo-preferences and prior.
  – At each step $i$
    • Make a greedy selection:
      
      $$\text{Sample } t^* \sim \nu^i, \ U^i \leftarrow \arg \max_{u \neq U^1, \ldots, U^{i-1}} g_{u^*t^*}$$

    • Observe the response $X^i$.
    • Update the pseudo-posterior: $\nu^{i+1}_t, \forall t \in T$.
  – Stop if:
    $$\nu^{i+1}_t \{ t : \exists u \neq U^1, \ldots, U^{i-1}, g_{ut} = 1 \} \leq \epsilon$$
Example

Groups of users, each prefers 1 topic

An additional topic is **pure spam**

When the truth is not spam, exploration has to continue

---

**Easier Case**

Groups of users, each prefers ½ of all topics

Topic identified “by bisection”, regret of order $\log |T|$
Summary

• Push-based news dissemination
• Extension of multi-armed bandit ideas
• **Greedy-Bayes** algorithm
  – *Simple yet provably order-optimal*
  – *Robust*, uses only partial preference information
  – *Adaptive* to structure
• Implications
  – Benchmark for push-pull decentralization
  – Importance of leveraging negative feedback
Open questions

• Is there a simple characterization in terms of \( \{p_{ut}\}, \ t \in T, \ u \in U \) of the optimal regret?

• Is Greedy-Bayes (or Thompson Sampling) order-optimal under more general conditions?

→ partial answers:
Toy examples where greed fails to probe highly informative nodes with high expected contribution to regret
Hope for somewhat general conditions under which Greedy Bayes performs near-bisection
Capacity of Information Processing Systems

- Systems that perform inference/categorization tasks, based on queries to resource-limited agents
  - Crowd sourcing → Human experts
  - Machine learning (e.g., clustering) → Cloud clusters
  - Medical lab diagnostics → Specialized lab machines

- Output based on large number of noisy inspections

Goal: identify efficient processing which uses minimum amount of resources
Problem Formulation

- Jobs arrive to system as unit rate Poisson process
- Job $i$ has hidden label $H_i$ in finite set $\mathcal{H}$
- $H_i$ generated i.i.d from distribution, $\pi$

$H_i \sim \pi$

$H_i = \text{‘cat’}$
(hidden)
Problem Formulation

• Goal: output estimated label, $H_i'$, s.t.

$$P(H_i' \neq h | H_i = h) \leq \delta, \quad \forall h$$
System resources

- $m$ experts of which $r_k$ fraction is type-$k$
- Type-$k$ expert when inspecting type $h$- job produces random response $X$:
  \[ P(X = x) = f(h, k, x) \]
  for distributions \{ $f(h, k, *)$ \} assumed known
- Single job inspection occupies an expert for time $\text{Exp}(1)$

\[ H_i' = \max_h \prod_{j=1}^{n} f(h, k_j, x_j) \]
Learning System

$H_i \sim \pi$

$H_i = 'cat'$ (hidden)

$P(\text{error}) \leq \delta$

$(x,1)$

$1 \quad 1 \quad 2 \quad 2 \quad 2 \quad K \quad K$

$m \cdot r_2$
Inspection Policy

• An inspection policy, $\phi$, decides

  1. Which experts to inspect which jobs
     - based on past history

  2. When a job will be allowed to depart from the system
Capacity Efficiency

- \( m_\phi (\pi, \delta) \) = smallest number of experts needed for stability under
  - policy \( \phi \)
  - job type distribution \( \pi \)
  - Estimation error \( \delta \)

- Let \( m^*(\pi, \delta) = \min_\phi m_\phi (\pi, \delta) \)

**Definition:** \( \phi \) is capacity efficient, if for all \( \pi \)

\[
\lim_{\delta \to 0} \frac{m_\phi (\pi, \delta)}{m^* (\pi, \delta)} = 1
\]
Main Result

Theorem

1. There exists a capacity efficient policy \( \phi \), s.t.

\[
\sup_{\pi} \frac{m_{\phi}(\pi, \delta)}{m^{*}(\pi, \delta)} \leq 1 + c \sqrt{\frac{\ln \ln(1/\delta)}{\ln(1/\delta)}} = 1 + \tilde{O}\left(\frac{1}{\sqrt{\ln(1/\delta)}}\right)
\]

2. The policy \( \phi \) does not require knowledge of \( \pi \)
Main Result (cont.)

• Makes no restrictions on response distributions \( \{f(h,k, *)\} \)

• Distribution \( \pi \) may be unknown and change over time

→ advantageous to have
  – Upper bound independent of \( \pi \)
  – \( \pi \)-oblivious inspection policy

• Constant \( c \) is explicit (but messy), and is proportional to
  – \( 1/ (\min_k r_k) \)
  – total number of labels
  – Ratio between KL-divergences among distributions of inspection results

\[
\sup_{\pi} \frac{m_{\phi}(\pi, \delta)}{m^*(\pi, \delta)} \leq 1 + c \sqrt{\frac{\ln \ln(1/\delta)}{\ln(1/\delta)}}
\]
Related Work

• Classical sequential hypothesis testing
  – Single expert by Wald 1945, Multi-expert model by Chernoff 1959
  – Costs measured by sample complexity
  – Does not involve finite resource constraint or simultaneous processing of multiple jobs

• Max-Weight policy in network resource allocation
  – Policy contains a variant of Max-Weight policy as a sub-module
  – Need to work with approximate labels and resolve synchronicity issues
Proof Outline

Main Steps:

1. Translate estimation accuracy into service requirements

2. Design processing architecture to balance contention for services among different job labels

Technical ingredients:

1. “Min-weight” adaptive policy using noisy job labels
2. Fluid model to prove stability over multiple stages
From Statistics to Services

• Consider single job $i$
• Suppose $i$ leaves system upon receiving $N$ inspections, with history

$$Y_N = \{(X_1, K_1), (X_2, K_2), \ldots (X_N, K_N)\}$$

• Write $f(h, y) = P(\text{seeing history } y|H_i = h)$
• Define likelihood ratio: $S_N(h, l) = \ln \frac{f(h, Y_N)}{f(l, Y_N)}$
Lemma (Sufficiency of a Likelihood Ratio Test)
Define event
\[ \mathcal{B} = \{ \exists H' \text{ s.t. } S_N(H', l) \geq \ln(|\mathcal{H}|/\delta), \quad \forall l \neq H' \} \]
Then
\[ \mathbf{P}(H' \neq h|\mathcal{B}, H_i = h) \leq \delta, \quad \forall h. \]

Lemma (Necessity)
If decision rule yields error $< \delta$ then it must be that
\[ \mathbf{E}(S_N(H_i, l)) \geq \ln(1/\delta), \quad \forall l \neq H_i \]
From Statistics to Services

• For true job type $H_i=h$, “service requirement” of job $i$:
  For all $l \neq h$ need to inspect job $i$ until $S(h,l)$ reaches $\ln(1/\delta)$

  $\rightarrow$ Job $i$ arrives with vector of “workloads” with initial level $= \ln(1/\delta)$ for all coordinates $l \neq h$

• Job of true type $h$ when inspected by type-$k$ expert receives “service” $\ln \left( \frac{f(h,k,X_j)}{f(l,k,X_j)} \right)$

  $\rightarrow$ Expected service amount: KL-divergence

$$D(h, l, k) = \mathbb{E}_h \left( \ln \frac{f(h,k,X)}{f(l,k,X)} \right)$$
Lower bound on performance

• If we knew true labels of all jobs and label distribution $\pi$, we would need $m$ as large as solution $\tilde{m}(\pi, \delta)$ of LP:

$$\text{minimize} \quad m$$

$$\text{s.t.} \quad \sum_h n_{h,k} \pi_h \leq r_k m, \quad \forall k,$$

$$\sum_k n_{h,k} D(h, l, k) \geq \ln(1/\delta), \quad \forall h \neq l$$

$$n_{h,k} \geq 0, \quad \forall h, k$$

Where $n_{h,k}$: Number of inspections of $h$-labeled job by type-$k$ expert

$\rightarrow$ Solution of LP, $\tilde{m}(\pi, \delta) = C \ln\left(\frac{1}{\delta}\right)$: lower bound on optimal capacity requirement
Processing Architecture

• Preparation stage: crude estimates of types from randomized inspections

• Adaptive stage: use crude estimates to “boot-strap” adaptive allocation policy for bulk of inspections and refinement of type estimates
  – Generate approximately optimal $n_{h,k}$ as in LP

• Residual stage: fix poorly learned labels from Adaptive Stage with randomized inspections
System size requirements

- **Preparation stage**
  - Short: $\approx \ln \ln \left( \frac{1}{\delta} \right)$ inspections
  - produces rough estimates

- **Adaptive stage**
  - Long: $\approx \ln \left( \frac{1}{\delta} \right) + \sqrt{\ln \left( \frac{1}{\delta} \right) \ln \ln \left( \frac{1}{\delta} \right)}$
  - produces good estimates with high probability

- **Residual stage**
  - Long $\approx \ln \left( \frac{1}{\delta} \right)$
  - produces good estimates $\frac{1}{\ln \left( \frac{1}{\delta} \right)}$
  - only invoked with small probability
System size requirements

• Total overhead compared to $m^*$

$$
\ln \ln(1/\delta) + \sqrt{\ln(1/\delta) \ln \ln(1/\delta)} + \ln(1/\delta) \ast \ln^{-1}(1/\delta)
$$

$$
= \tilde{O}(\sqrt{\ln(1/\delta)}) = \tilde{O} \left( \frac{1}{\sqrt{\ln(1/\delta)}} \right) m^*(\pi, \delta)
$$
Conclusions

• Considered resource allocation issues for “learning systems”

• Leads to unusual combination of sequential hypothesis testing with network resource scheduling

• Outlook
  – Simpler solutions with better performance?
  – Impact of number of types, and structure of expertise?