Introduction to Active Learning

Nandana Sengupta

November 9, 2015

Active Learning: Introduction

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:

Active Learning: Introduction

- “n” objects: $\theta_1, \theta_2, \cdots \theta_n$.
- “m” individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the “n” objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.
  - matrix of individual rankings

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.
  - matrix of individual rankings
  - eg: Netflix problem.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.
  - matrix of individual rankings
  - eg: Netflix problem.
- Issues in generating such rankings: costly queries, missing value problem, inefficiencies.

Active Learning: Introduction

- "n" objects: $\theta_1, \theta_2, \cdots \theta_n$.
- "m" individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the "n" objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.
  - matrix of individual rankings
  - eg: Netflix problem.
- Issues in generating such rankings: costly queries, missing value problem, inefficiencies.
- Typically random queries over "n" objects $\Rightarrow$ number of queries: $\binom{n}{2}$

Active Learning: Introduction

- “n” objects: $\theta_1, \theta_2, \cdots \theta_n$.
- “m” individuals: $i_1, i_2, \cdots i_m$.
- Aim is to get a ranking over the “n” objects:
  - global ranking: $\theta_2 \succ \theta_3 \succ \theta_1 \succ \theta_5 \succ \theta_6 \succ \theta_4$.
  - eg: web search queries, street safety queries.
  - matrix of individual rankings
  - eg: Netflix problem.
- Issues in generating such rankings: costly queries, missing value problem, inefficiencies.
- Typically random queries over “n” objects $\Rightarrow$ number of queries: $\binom{n}{2}$
- Inefficient! This is where Active Learning comes in.

Active Learning: Introduction

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.
  - However r is unknown – so problem becomes estimating region where r lies.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.
  - However r is unknown – so problem becomes estimating region where r lies.
  - Some queries redundant – example next.

Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.
  - However r is unknown – so problem becomes estimating region where r lies.
  - Some queries redundant – example next.
- Assumptions

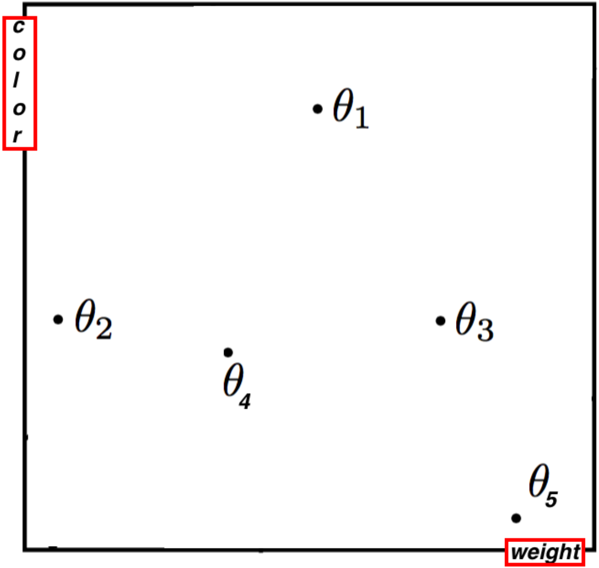
Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.
  - However r is unknown – so problem becomes estimating region where r lies.
  - Some queries redundant – example next.
- Assumptions
  - A1: Embedding if $\theta_i \prec \theta_j$ then $\|\theta_i - r\| < \|\theta_j - r\|$.

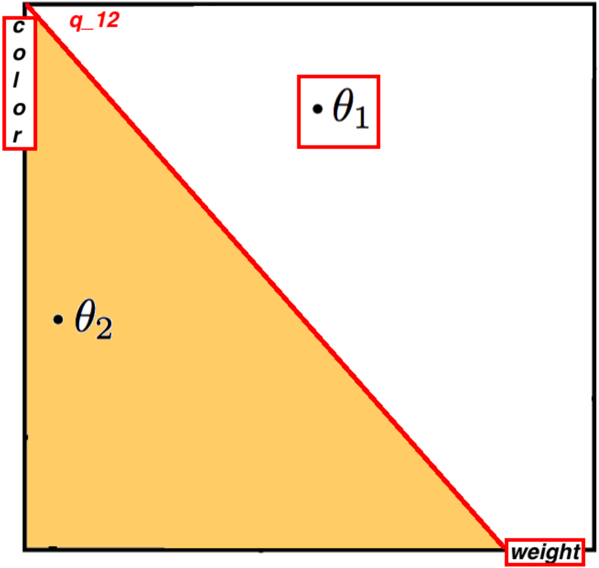
Active Learning: Introduction

- "Active Ranking using Pairwise Comparisons" – Jamieson and Nowak (2011)
- if there exists $d < n$ features in $\mathcal{R}^d$ then can reduce order of queries: $\mathcal{O}(d \log n)$.
- Main idea is to not query randomly but use these d features.
  - Reference Point in "d"-dimensional space, say r.
  - Distance from r determines ranking.
  - However r is unknown – so problem becomes estimating region where r lies.
  - Some queries redundant – example next.
- Assumptions
  - A1: Embedding if $\theta_i \prec \theta_j$ then $\|\theta_i - r\| < \|\theta_j - r\|$.
  - A2: Consistency Every pairwise comparison is consistent with the global ranking.

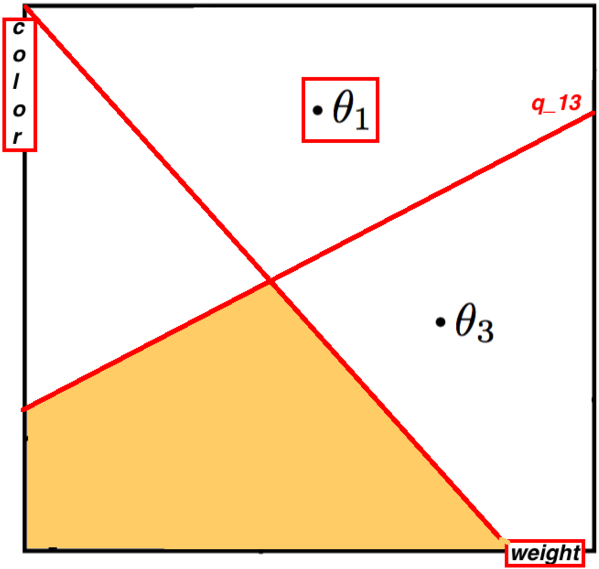
Active Learning: Example



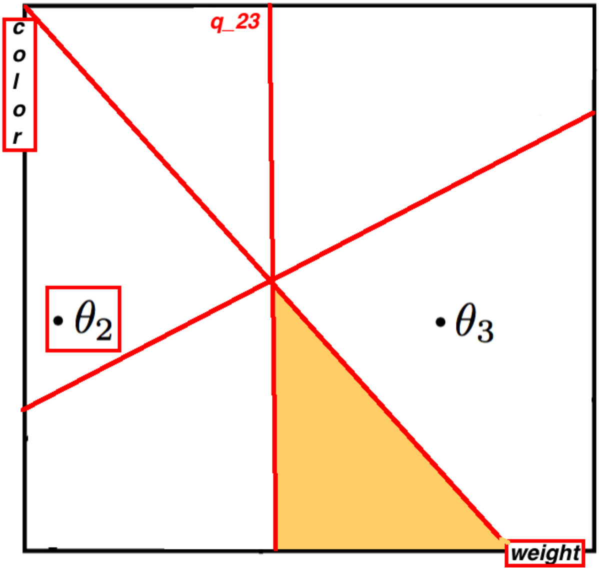
Active Learning: Example



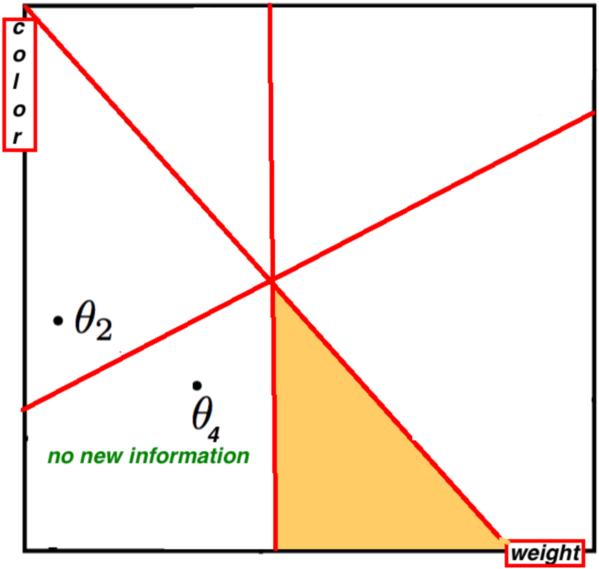
Active Learning: Example



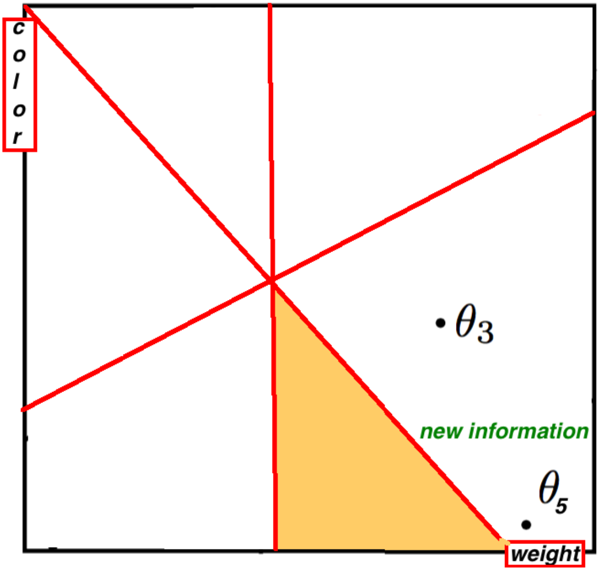
Active Learning: Example



Active Learning: Example



Active Learning: Example



Active Learning: Possible Applications

Active Learning: Possible Applications

- Pairwise ranking surveys online.

Active Learning: Possible Applications

- Pairwise ranking surveys online.
- Eg: Street Score, All our ideas.

Active Learning: Possible Applications

- Pairwise ranking surveys online.
- Eg: Street Score, All our ideas.
- Generate global rankings by asking pairwise comparison queries.

Active Learning: Possible Applications

- Pairwise ranking surveys online.
- Eg: Street Score, All our ideas.
- Generate global rankings by asking pairwise comparison queries.
- Currently using random pairwise queries.

Active Learning: Possible Applications

- Pairwise ranking surveys online.
- Eg: Street Score, All our ideas.
- Generate global rankings by asking pairwise comparison queries.
- Currently using random pairwise queries.
- Opportunity for active learning algorithms.

Thanks!