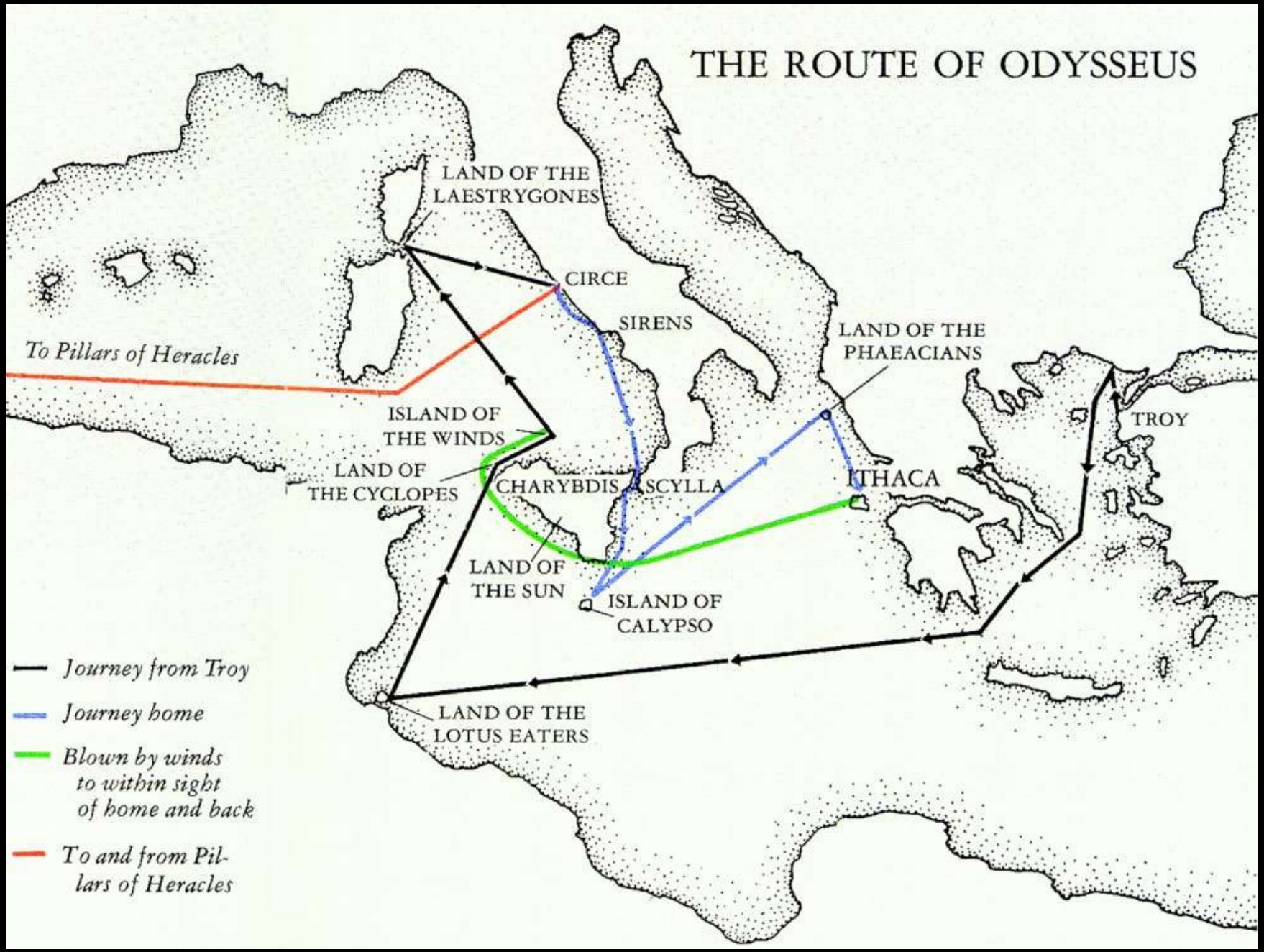
An Optimization Framework for Query Recommendation

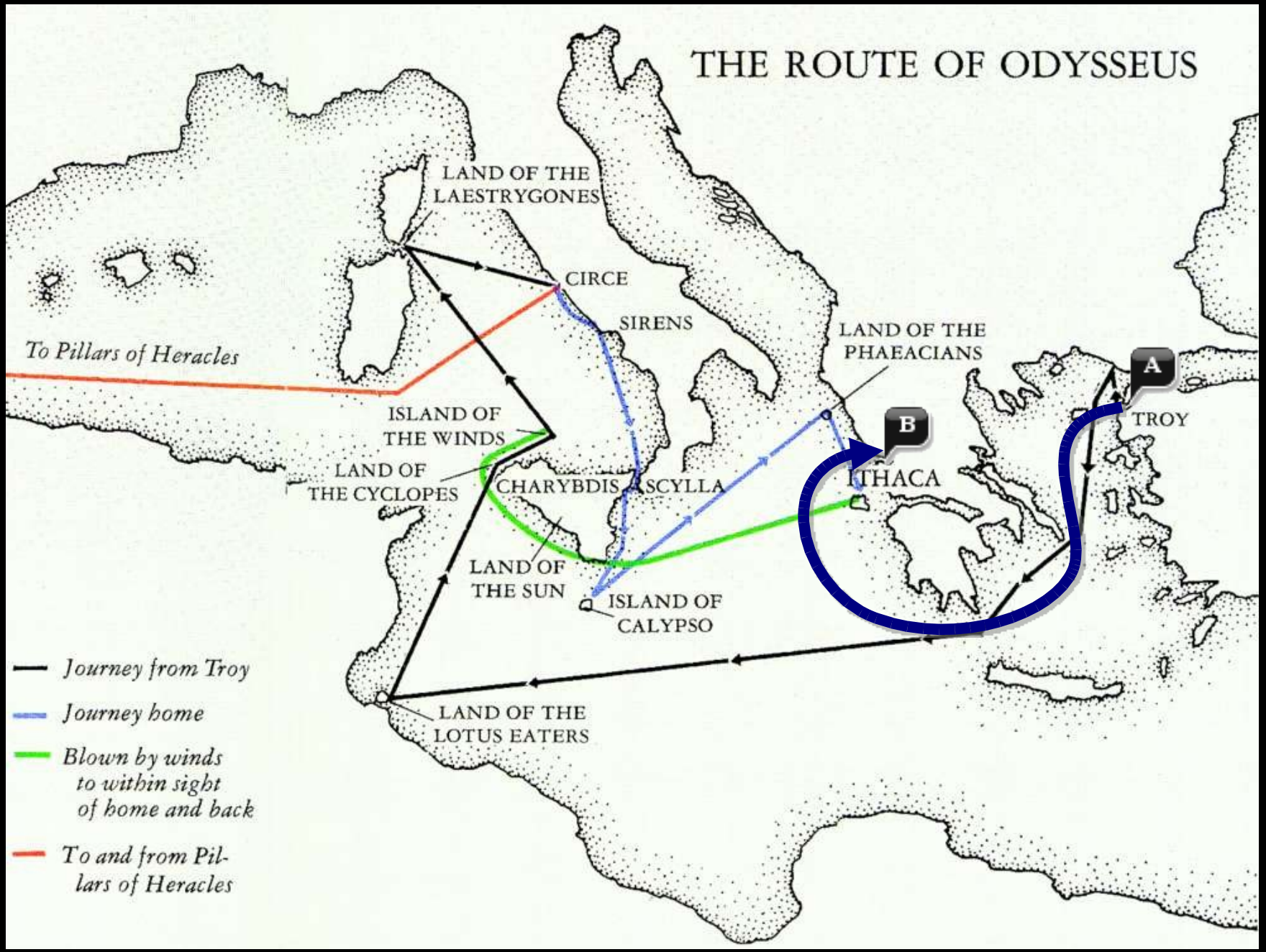
Aris Anagnostopoulos, Luca Becchetti,

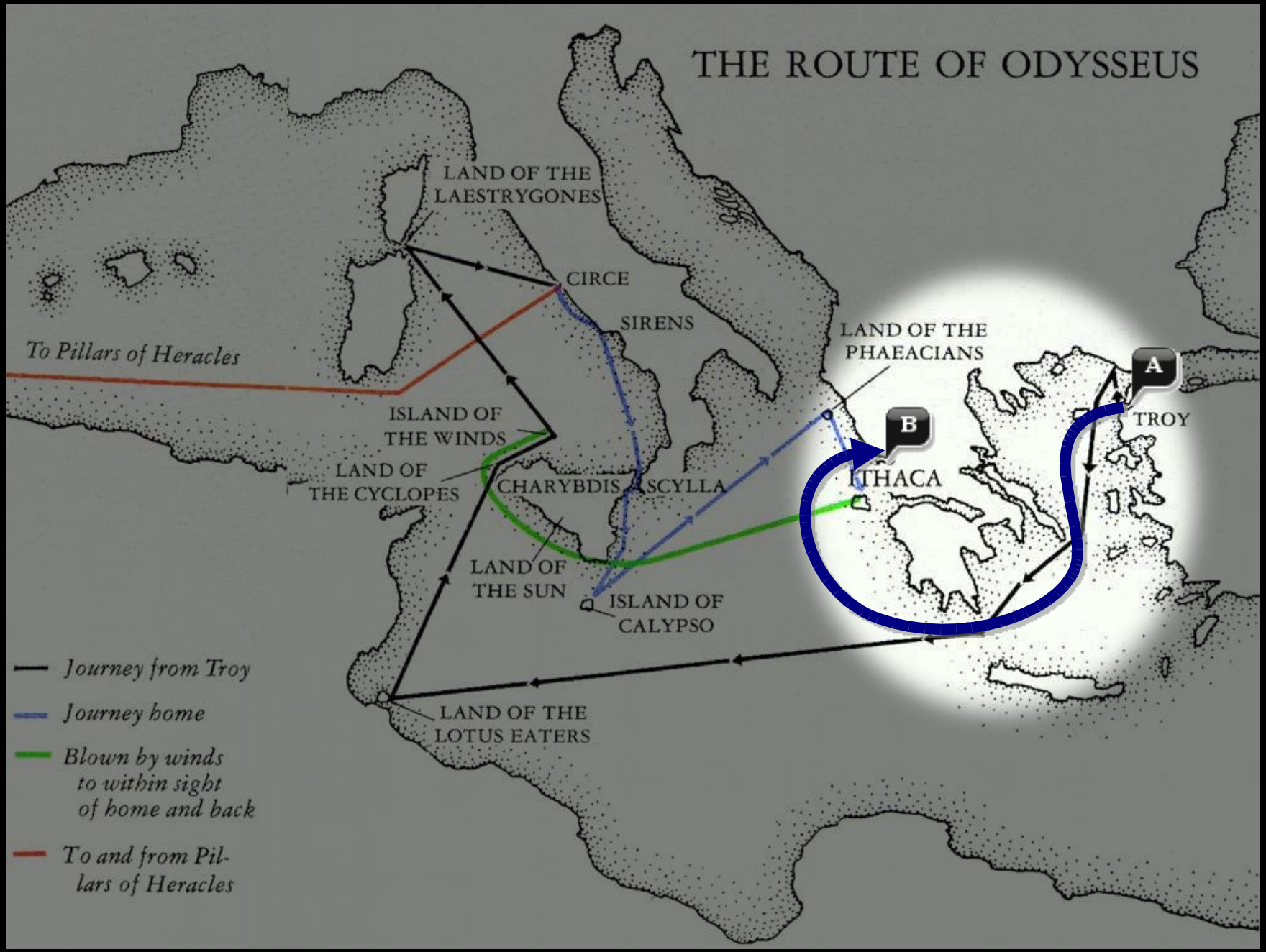
Carlos Castillo, Aristides Gionis

YAHOO! RESEARCH



Source: B. Precourt at UWM.


THE ROUTE OF ODYSSEUS
LAND OF THE LAESTRYGONES
CIRCE
SIRENS
LAND OF THE PHAEACIANS
To Pillars of Heracles
A
TROY
B
ITHACA
ISLAND OF THE WINDS
LAND OF THE CYCLOPES
CHARYBDIS
SCYLLA
LAND OF THE SUN
ISLAND OF CALYPSO
LAND OF THE LOTUS EATERS
Journey from Troy
Journey home
Blown by winds to within sight of home and back
To and from Pillars of Heracles


THE ROUTE OF ODYSSEUS
LAND OF THE LAESTRYGONES
CIRCE
SIRENS
To Pillars of Heracles
LAND OF THE PHAEACIANS
A
TROY
B
ITHACA
ISLAND OF THE WINDS
LAND OF THE CYCLOPES
CHARYBDIS
SCYLLA
LAND OF THE SUN
ISLAND OF CALYPSO
LAND OF THE LOTUS EATERS
Journey from Troy
Journey home
Blown by winds to within sight of home and back
To and from Pillars of Heracles

As you set

out for Ithaca,

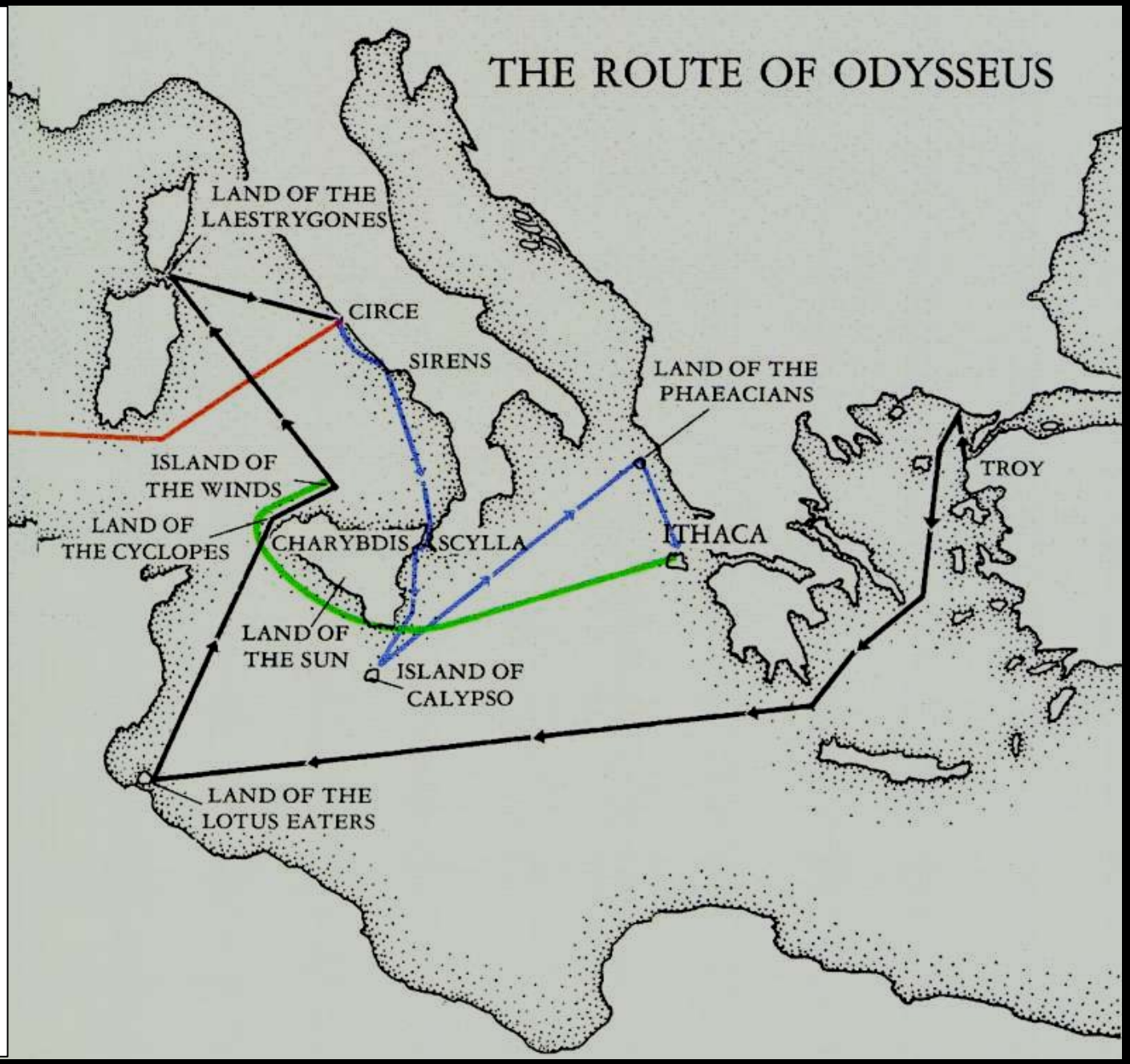
hope your road

is a long one,

full of adventure,

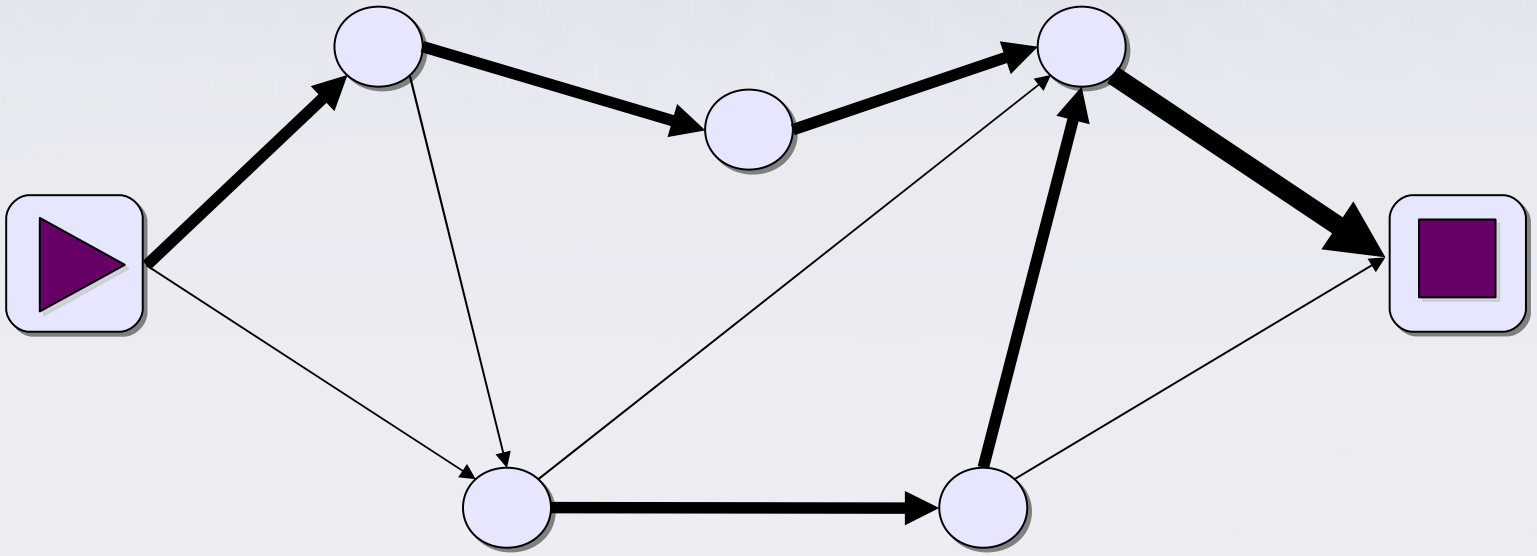
full of discovery.

K. Kavafis



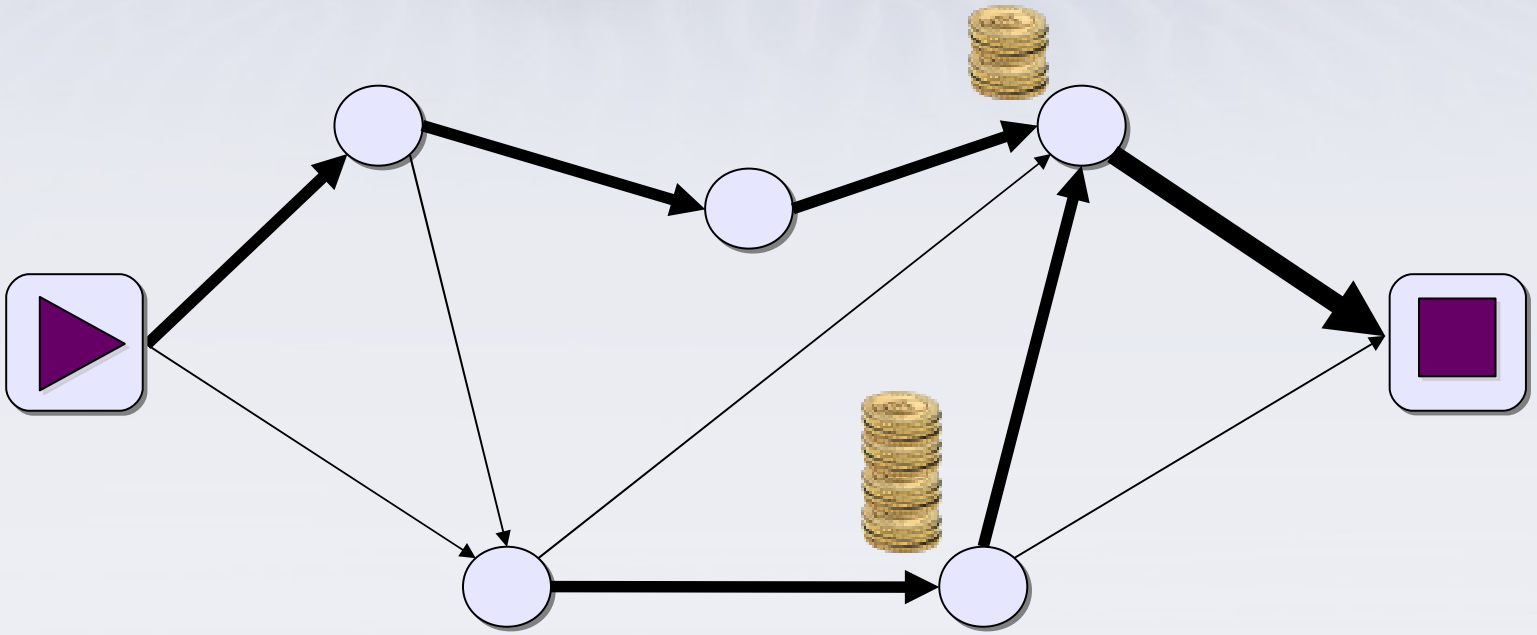
Problem

Given a set of possible user histories



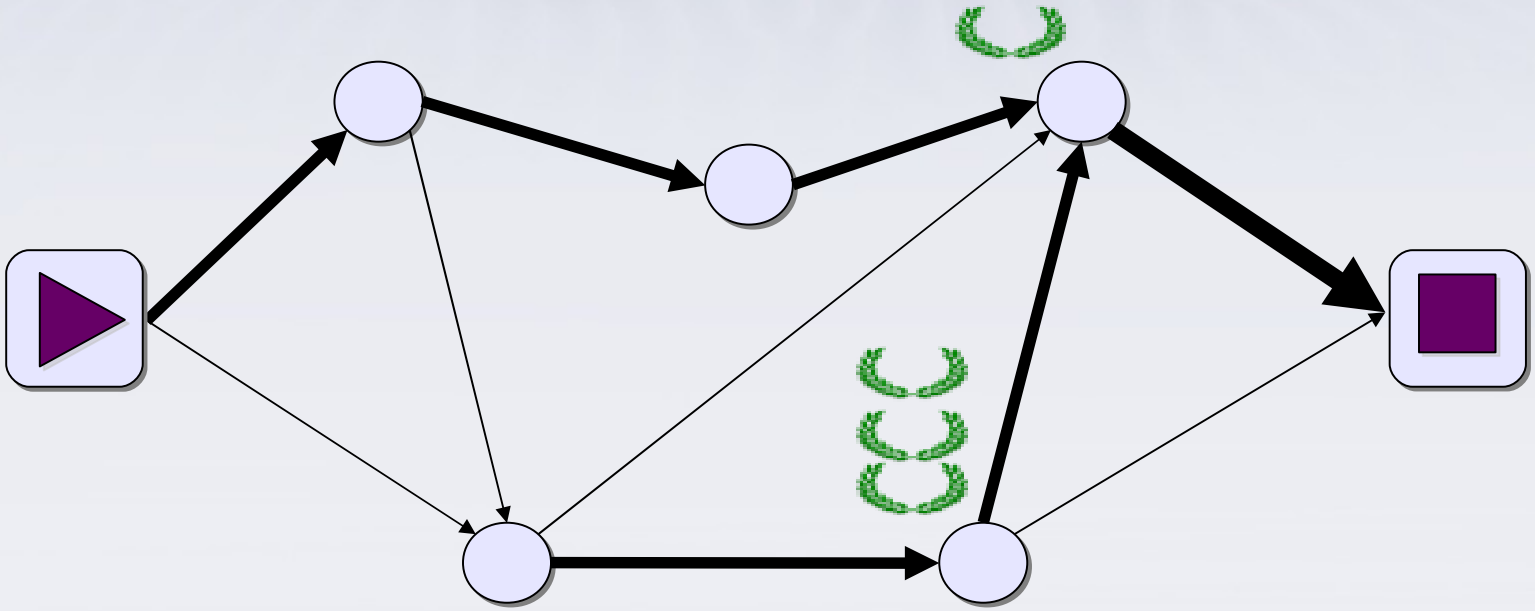
Problem

Given a value for different states



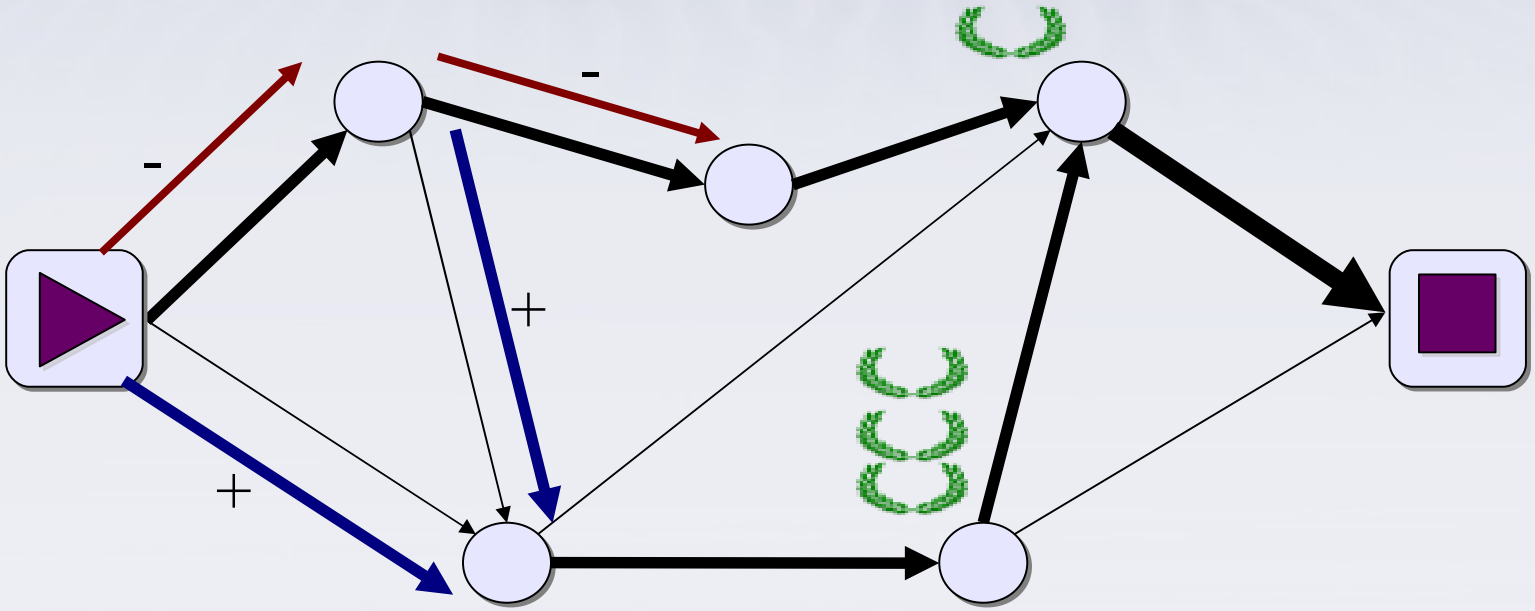
Problem

Given a value for different states



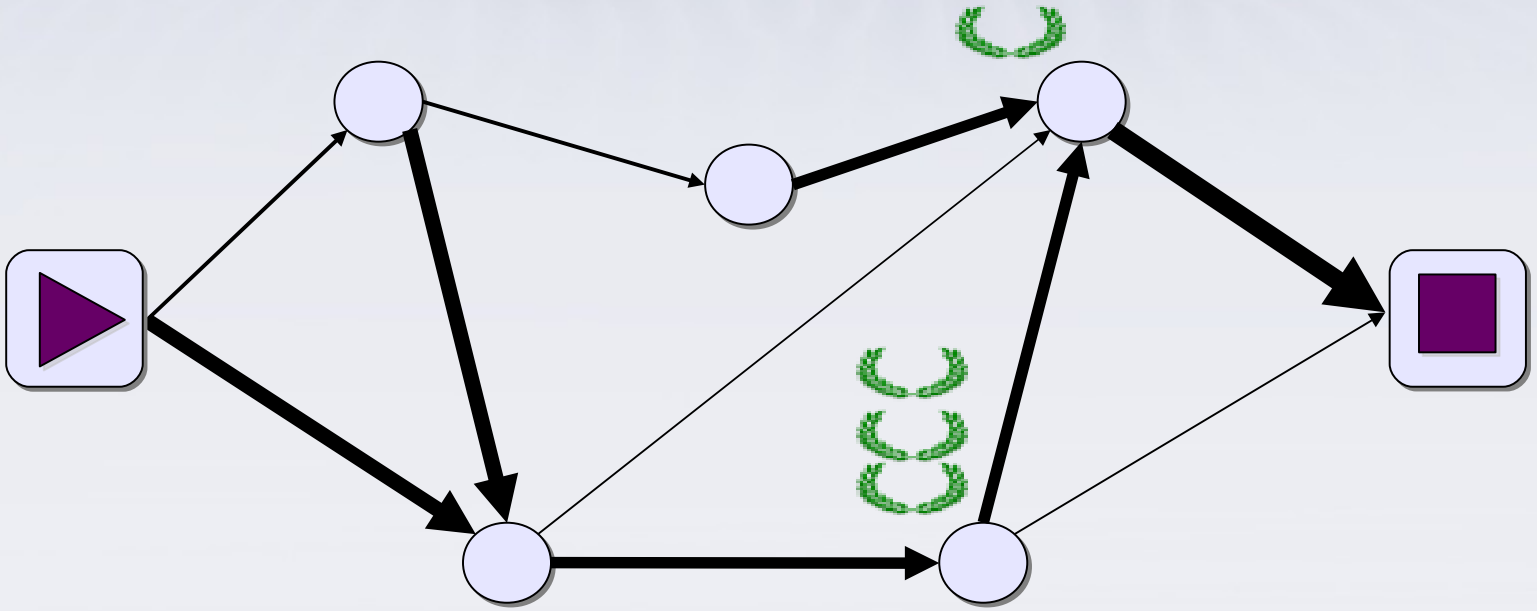
Problem

"Nudge" the users in a certain direction



Problem

"Nudge" the users in a certain direction



Problem definition

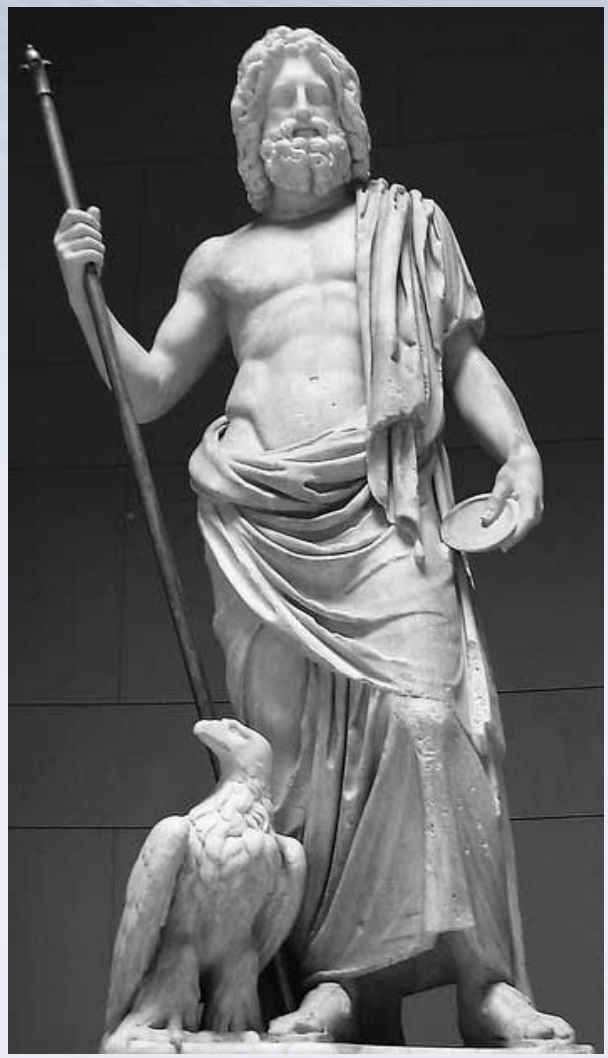
- Given a set of possible user sessions
- Given a certain value for different states
- "Nudge" the users in a certain direction

Objectives

  - 1. collect a large reward along the way -or-
  - 2. end the session at a rewarding action

Constraints

- We are not almighty



Source: not-of-this-earth.com

Constraints

- We are not almighty
  - We can only suggest, not order

Constraints

- We are not almighty
  - We can only suggest, not order
- We are not all-knowing

Source: Wikimedia commons.

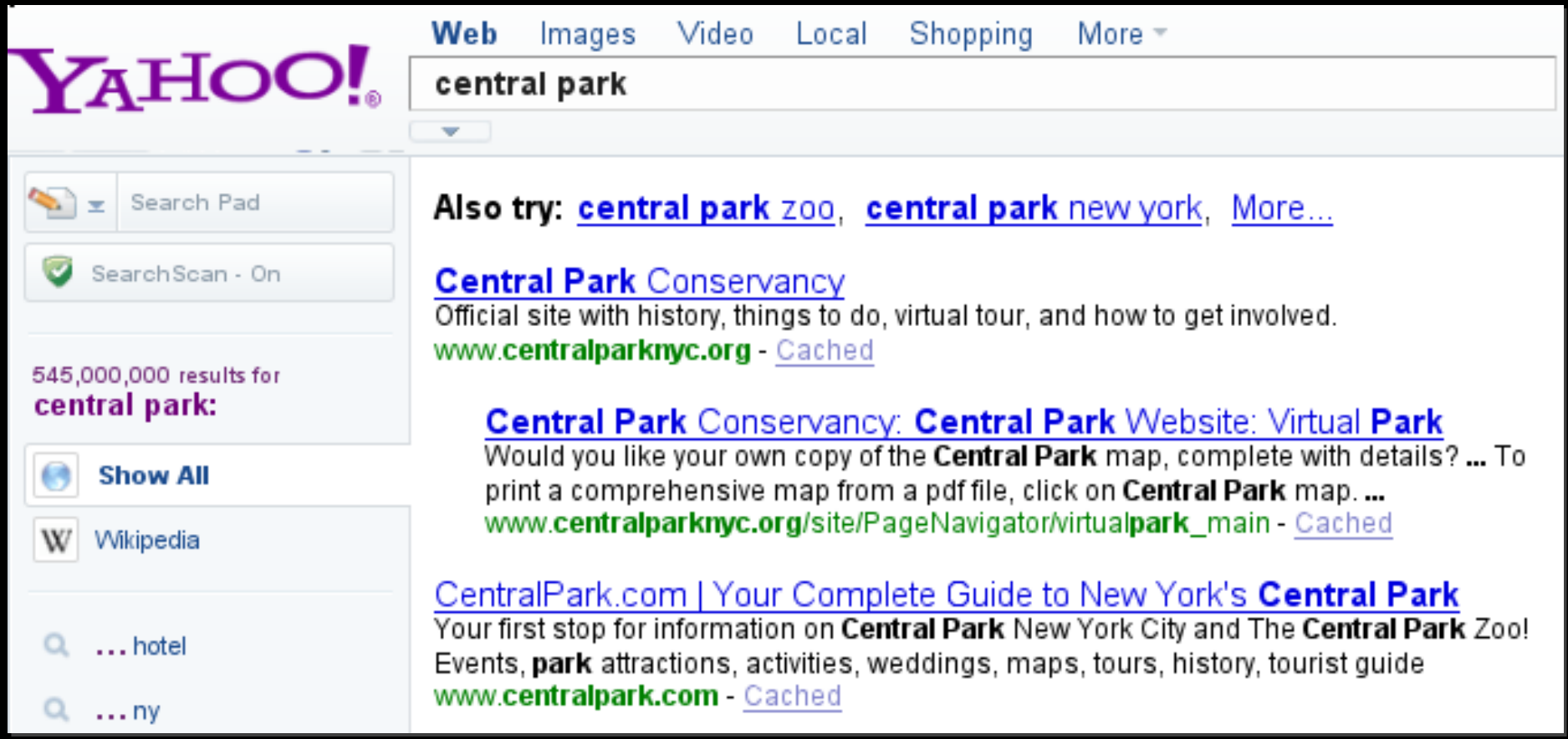
Constraints

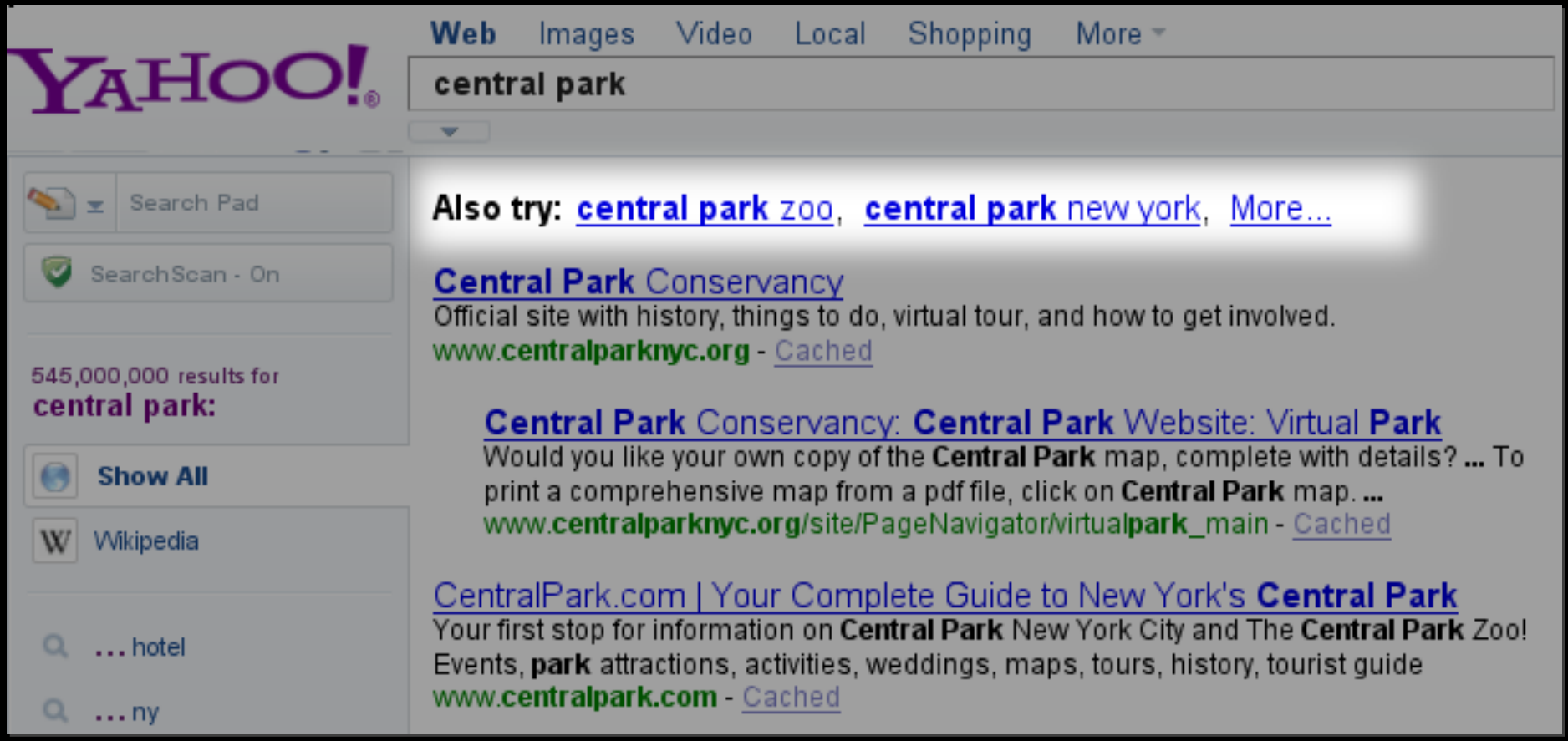
- We are not almighty
  - We can only suggest, not order
- We are not all-knowing
  - We do not know how the users will react

Framework

Setting

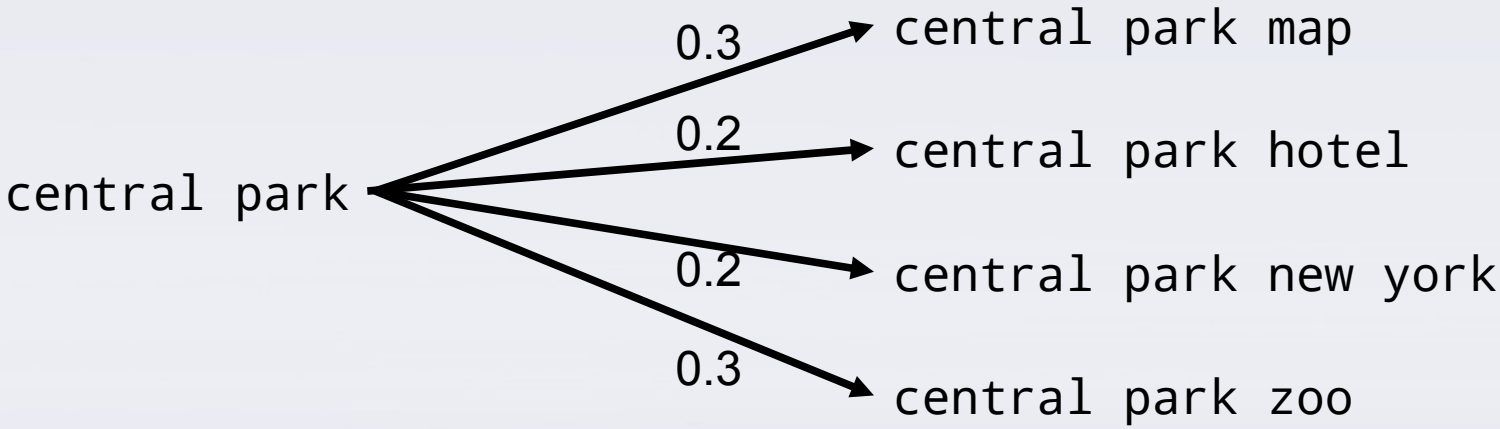
- Paper: general framework
  - e.g. for optimizing links on web sites
- Talk: query recommendation


YAHOO!
Web Images Video Local Shopping More
central park
Search Pad
SearchScan - On
545,000,000 results for
central park:
Show All
Wikipedia
... hotel
... ny
Also try: central park zoo, central park new york, More...
Central Park Conservancy
Official site with history, things to do, virtual tour, and how to get involved.
www.centralparknyc.org - Cached
Central Park Conservancy: Central Park Website: Virtual Park
Would you like your own copy of the Central Park map, complete with details? ... To print a comprehensive map from a pdf file, click on Central Park map. ...
www.centralparknyc.org/site/PageNavigator/virtualpark_main - Cached
CentralPark.com | Your Complete Guide to New York's Central Park
Your first stop for information on Central Park New York City and The Central Park Zoo! Events, park attractions, activities, weddings, maps, tours, history, tourist guide
www.centralpark.com - Cached


YAHOO!
Web
Images
Video
Local
Shopping
More
central park
Search Pad
SearchScan - On
545,000,000 results for
central park:
Show All
Wikipedia
... hotel
... ny
Also try: central park zoo, central park new york, More...
Central Park Conservancy
Official site with history, things to do, virtual tour, and how to get involved.
www.centralparknyc.org - Cached
Central Park Conservancy: Central Park Website: Virtual Park
Would you like your own copy of the Central Park map, complete with details? ... To print a comprehensive map from a pdf file, click on Central Park map. ...
www.centralparknyc.org/site/PageNavigator/virtualpark_main - Cached
CentralPark.com | Your Complete Guide to New York's Central Park
Your first stop for information on Central Park New York City and The Central Park Zoo! Events, park attractions, activities, weddings, maps, tours, history, tourist guide
www.centralpark.com - Cached

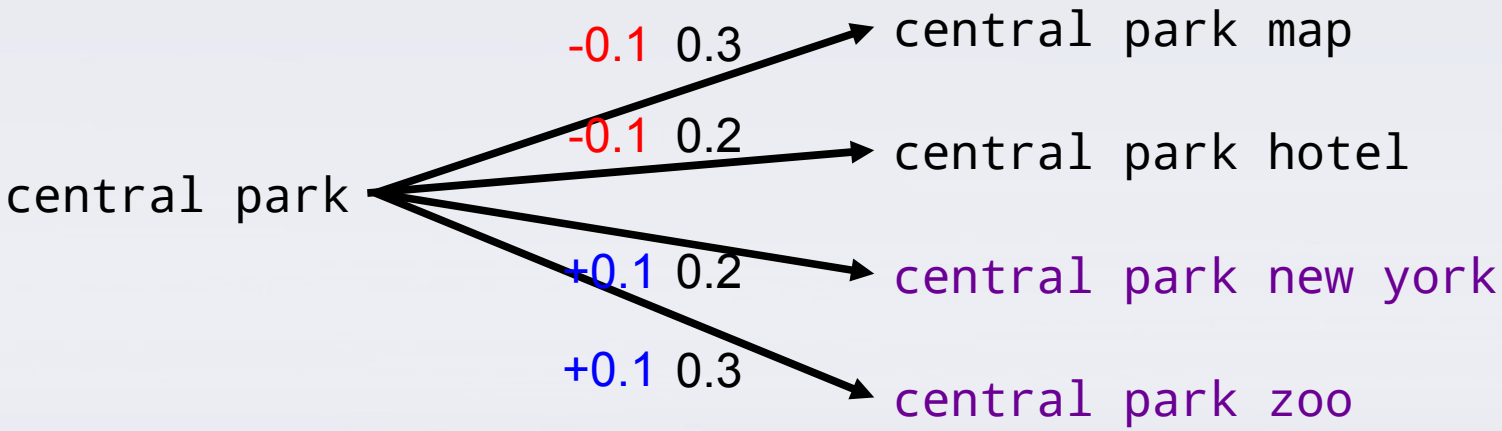
Query recommendations

- Reformulation probabilities
  - P(q,q') original



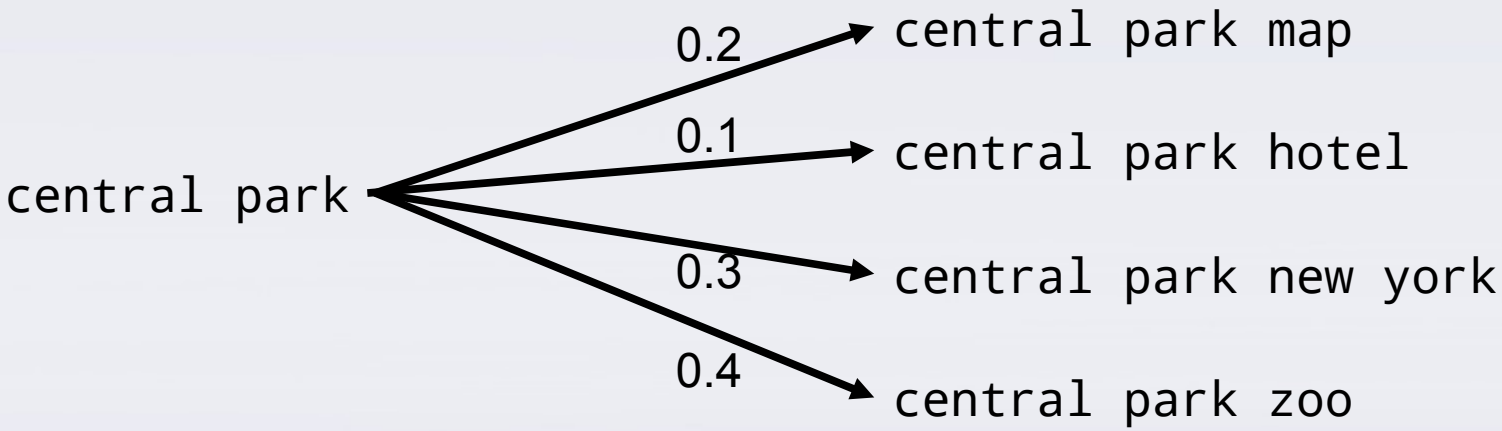
Query recommendations

- Reformulation probabilities
  - P(q,q') original
  - P'(q,q') perturbed = P(q,q') + ρ(Q,q,q')



Query recommendations

- Reformulation probabilities
  - P(q,q') original
  - P'(q,q') perturbed = P(q,q') + ρ(Q,q,q')



Example values w(·)

- Search engine results page
  - Quality of search results
- General page
  - Dwell time
  - User ratings

Objective functions U(·)

Kavafian

"a road full of adventure"

$$U(\langle q_1, q_2, \ldots, q_t \rangle) = \sum w(q_i)$$

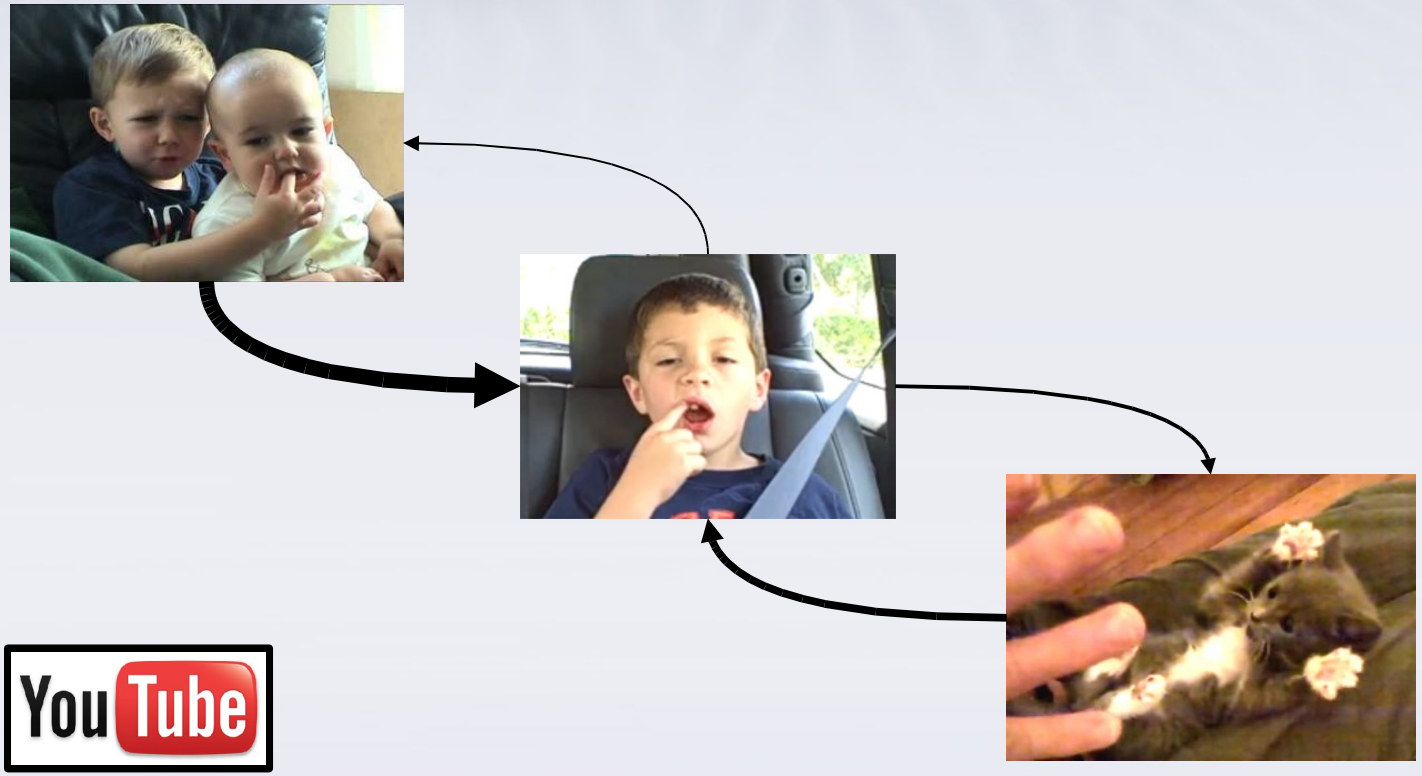
Machiavellian

"ends justify means"

$$U(\langle q_1, q_2, \ldots, q_t \rangle) = w(q_t)$$

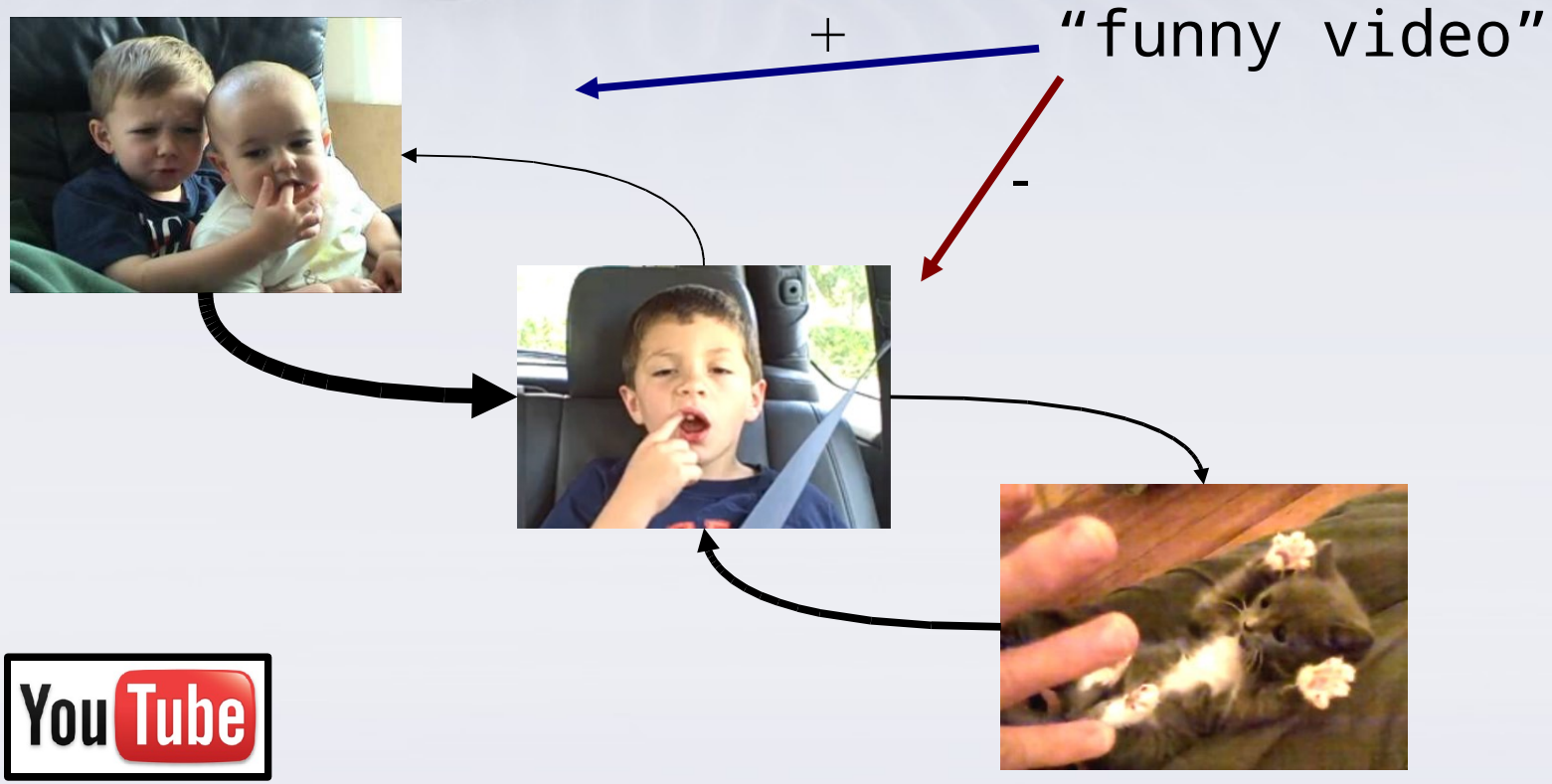
The *Kavafian* objective

Useful when users want to explore, or be entertained



The *Kavafian* objective

Useful when users want to explore, or be entertained

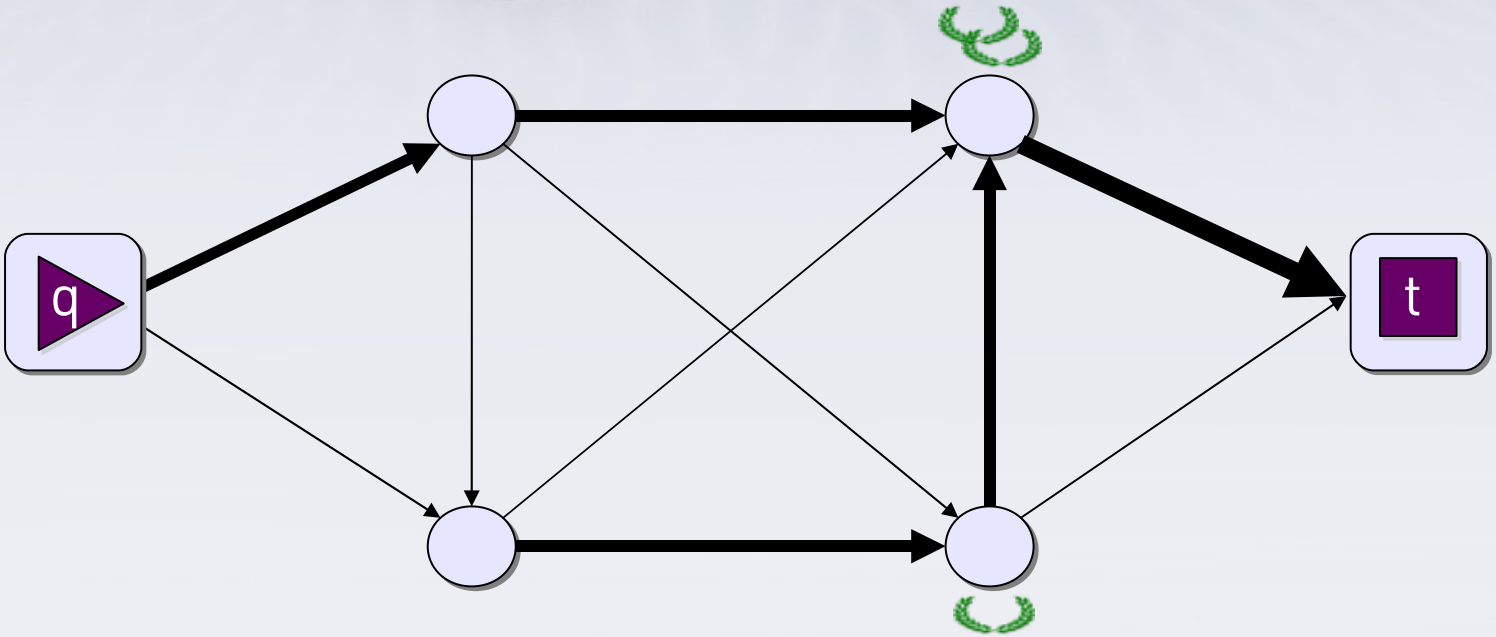


Optimization problem

- Given:
  - Original transition probabilities P
  - Starting node q
  - Node values w
  - Perturbation function ρ
- Add up to k links (per node), maximize expected utility of paths starting at q

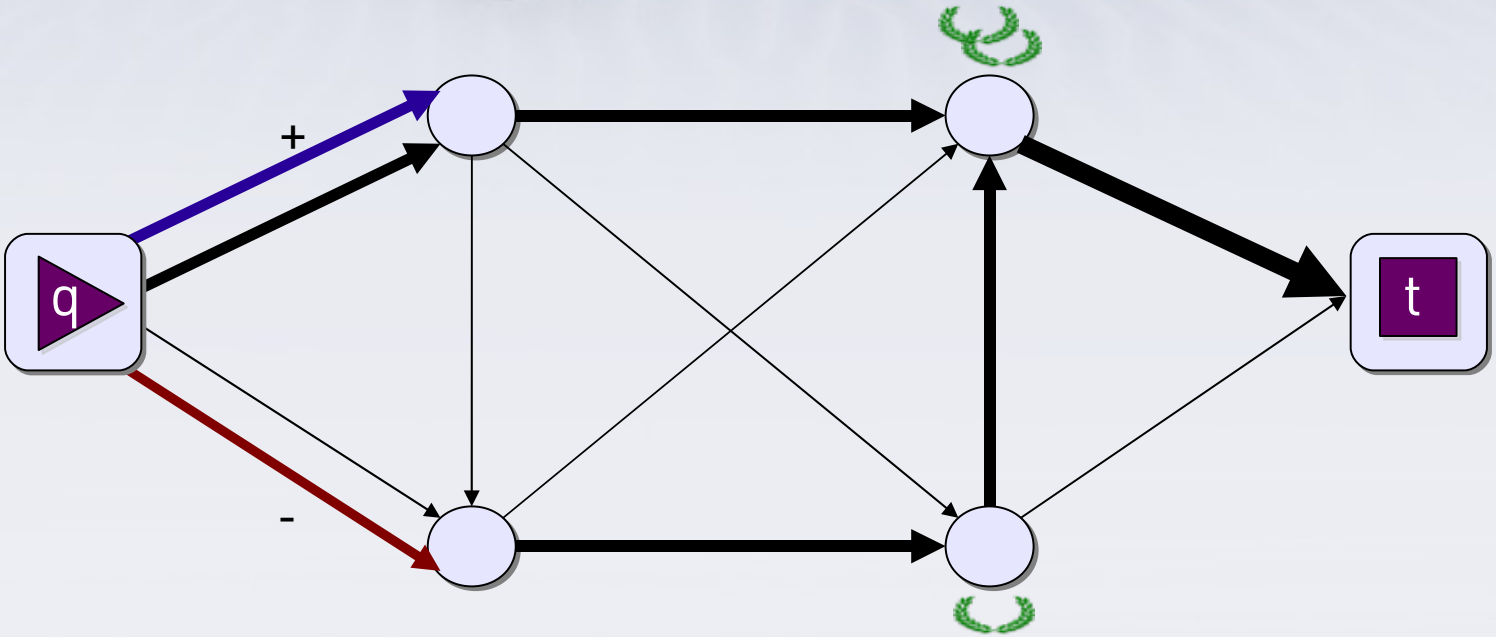
Single-step recommendation

- Recommend now, leave user alone later



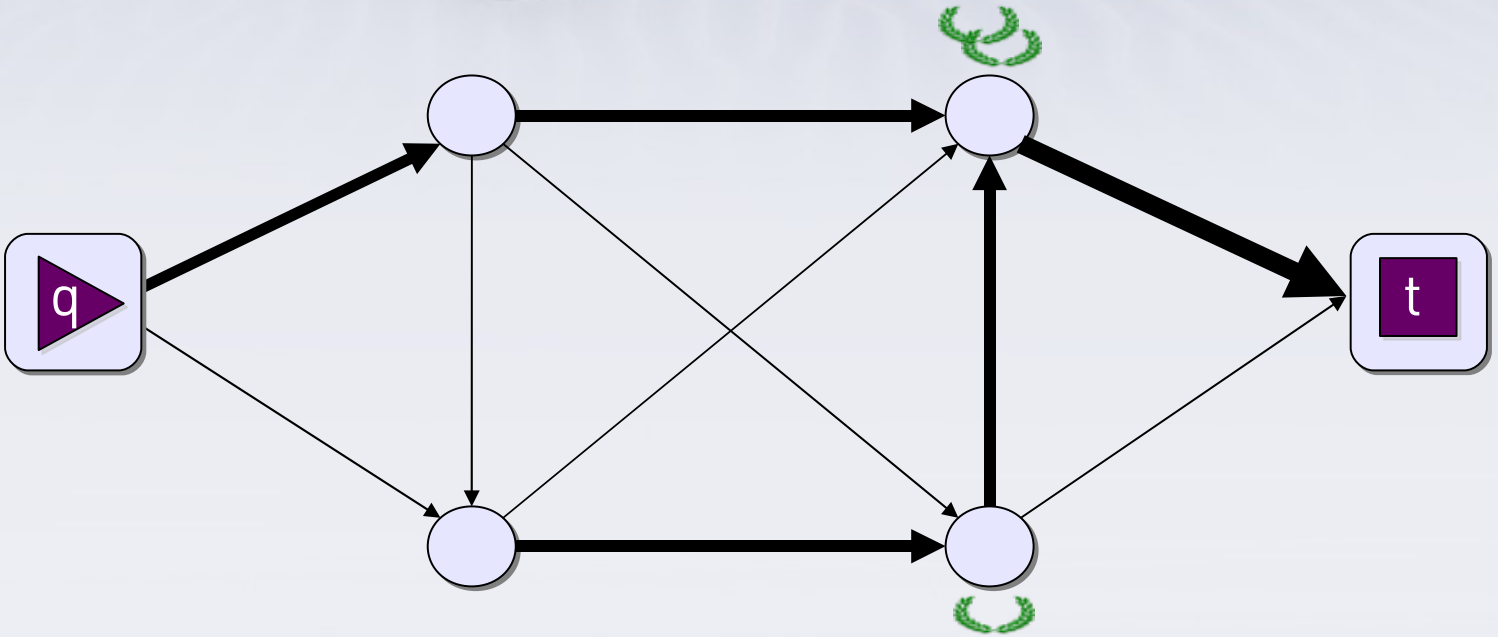
Single-step recommendation

- Recommend now, leave user alone later



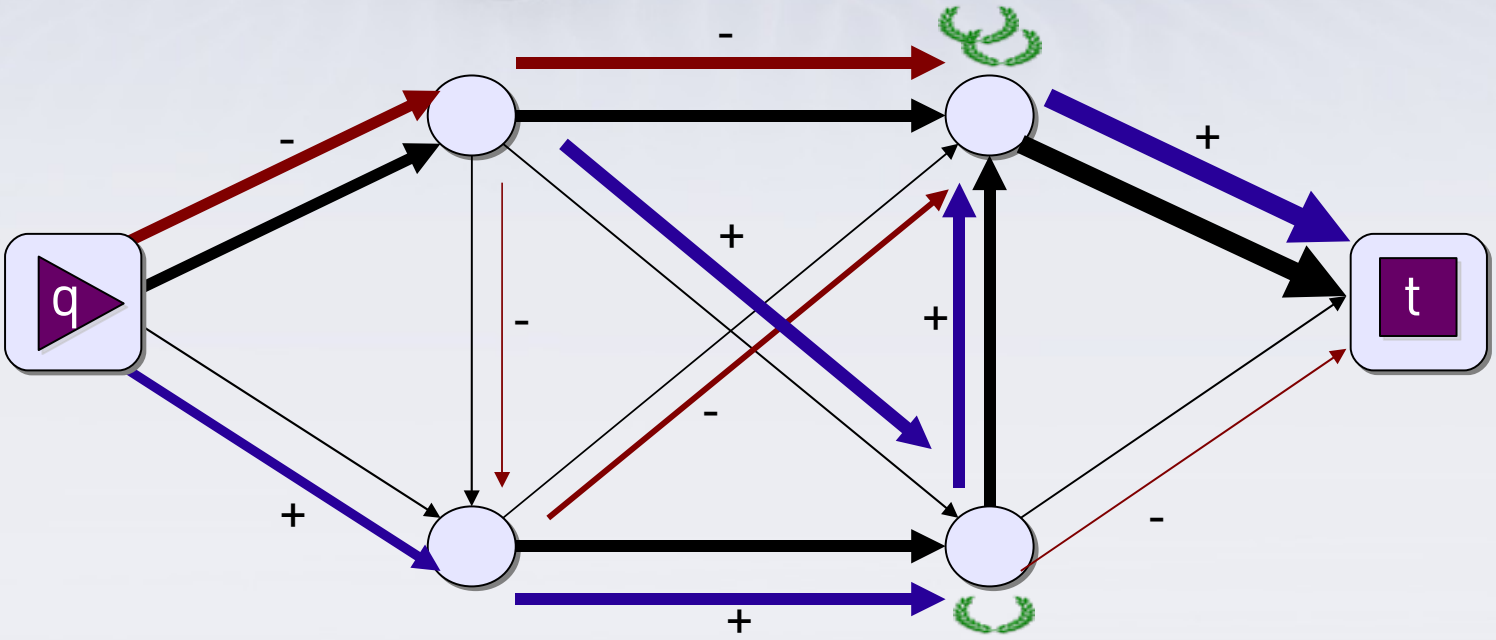
Multi-step recommendation

- Recommend at each step in the future



Multi-step recommendation

- Recommend at each step in the future



Multi-step case

- **Multi-** and **Single-step** recommendation are NP-complete
  - Reduction from MAXIMUM-COVER
- Heuristic for multi-step problem:
  - at each node, assume rest of the graph is unperturbed when computing utility of adding an edge

Single-step case

- Greedy heuristic for *"Machiavellian"* objective: find (q_i, q_j) maximizing

$$\rho_{ij}(E[U(path(q_j))] - w_i)$$

- Repeat k times

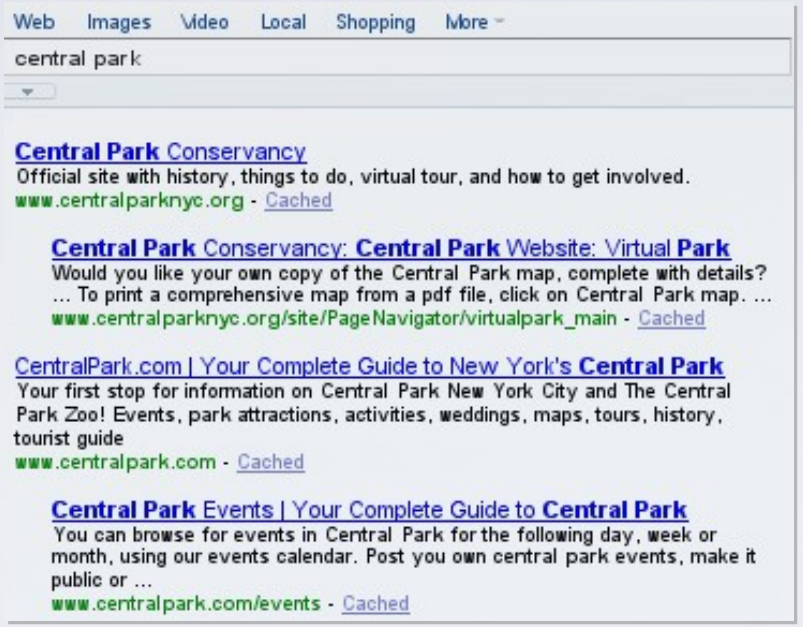
Observation

- Greedy heuristic achieves utility at least (1-x) of the optimum, with x << 1 in cases of practical interest
  - It is possible to construct pathological instances s.t. that greedy performs poorly
  - x depends on termination probability at q_i and probabilities of following recommendations

Application

Large-scale experiment

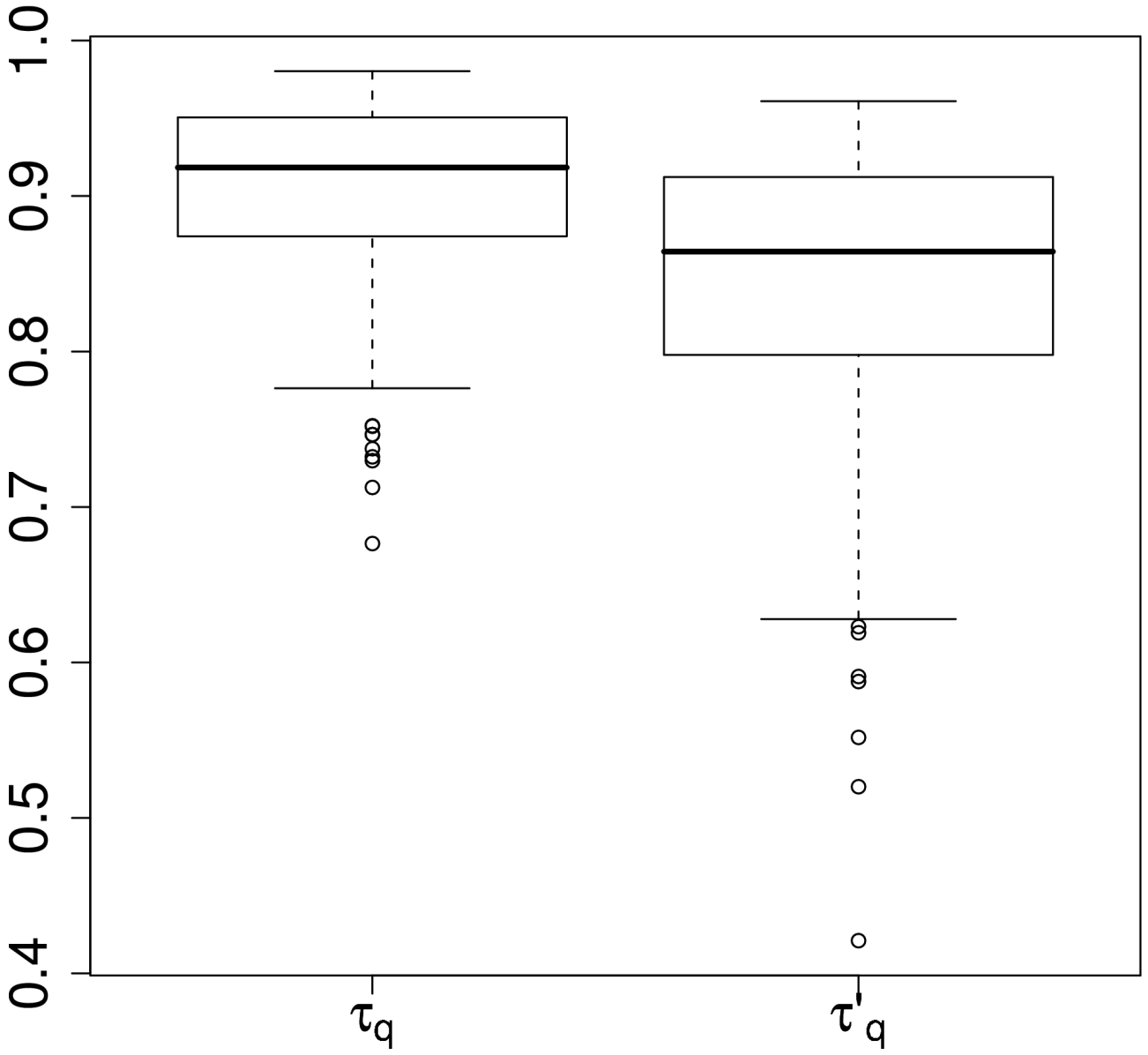
We observe perturbed probabilities P'(q,q'), unless we disable search assist to see P(q,q')



Empirical observations

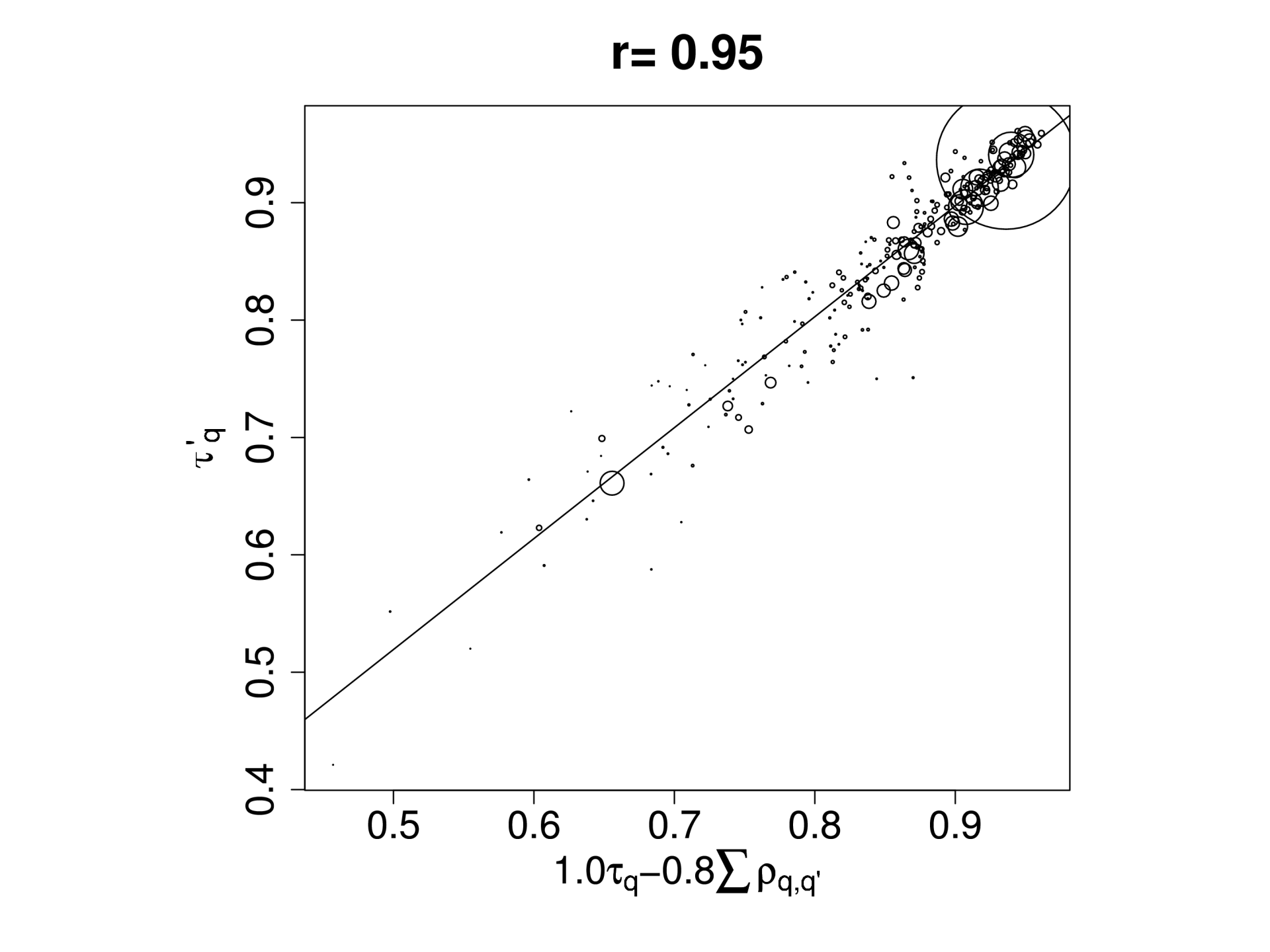
- Notation:
  - τ_q, τ'_q session-end probabilities
- Recommendations decrease termination probability:
  - Average $\tau_q \approx 0.90$
  - Average $\tau'_q \approx 0.84$

Termination probability



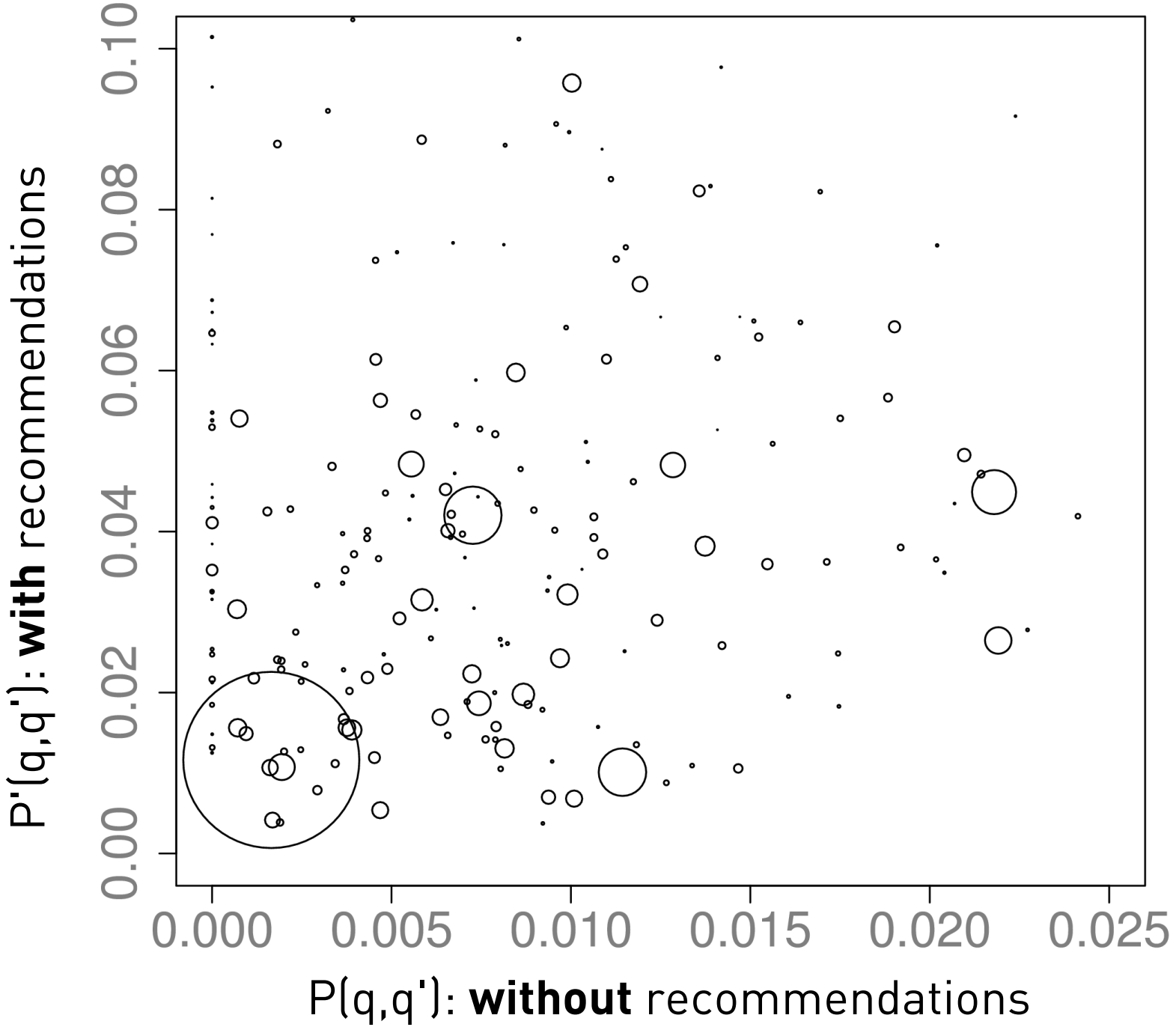
Empirical observations

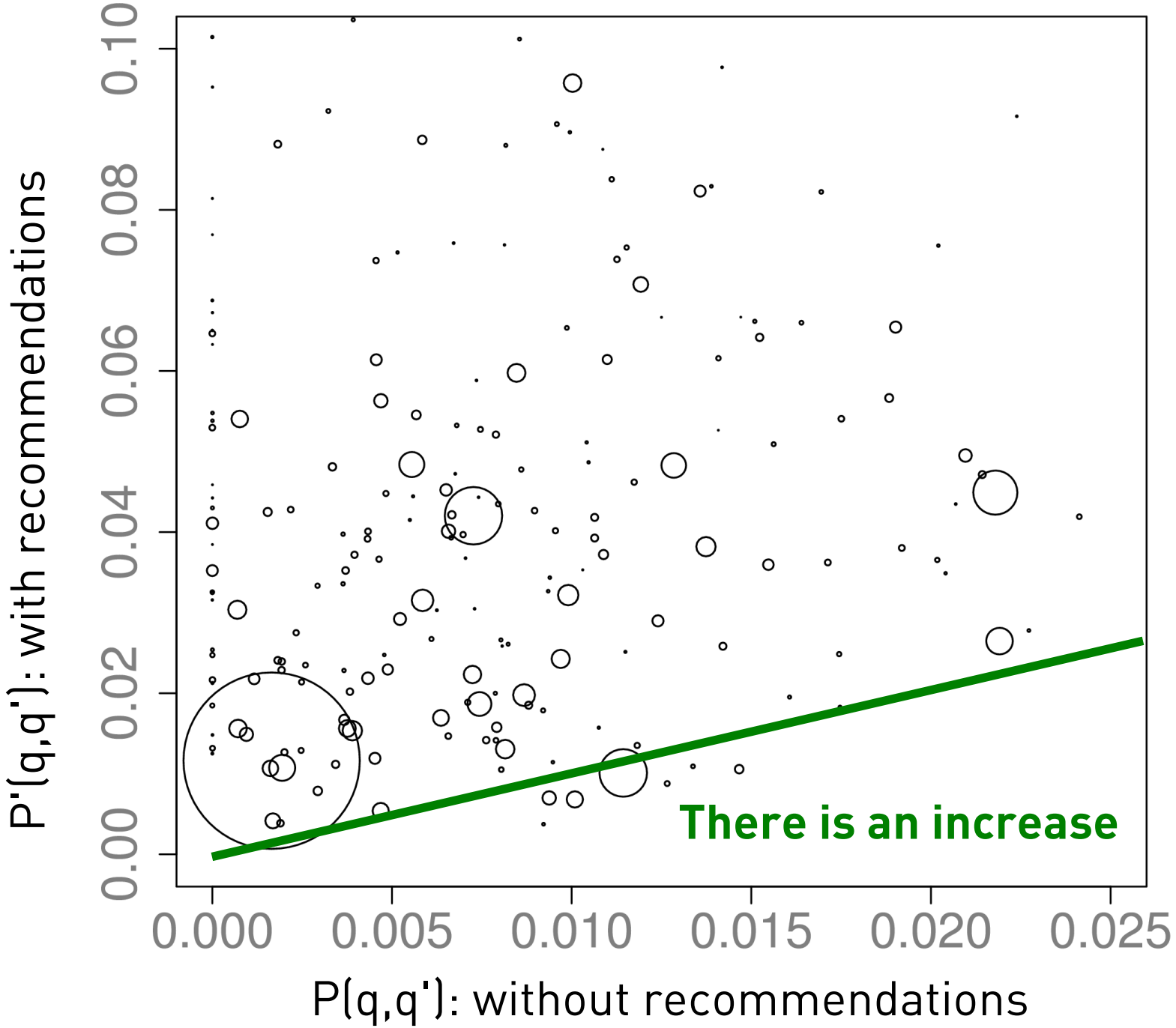
- Recommendations decrease termination probability, $\tau_q \approx 0.90$ $\tau'_q \approx 0.84$
- Decrease is almost entirely due to more clicks on recommendations

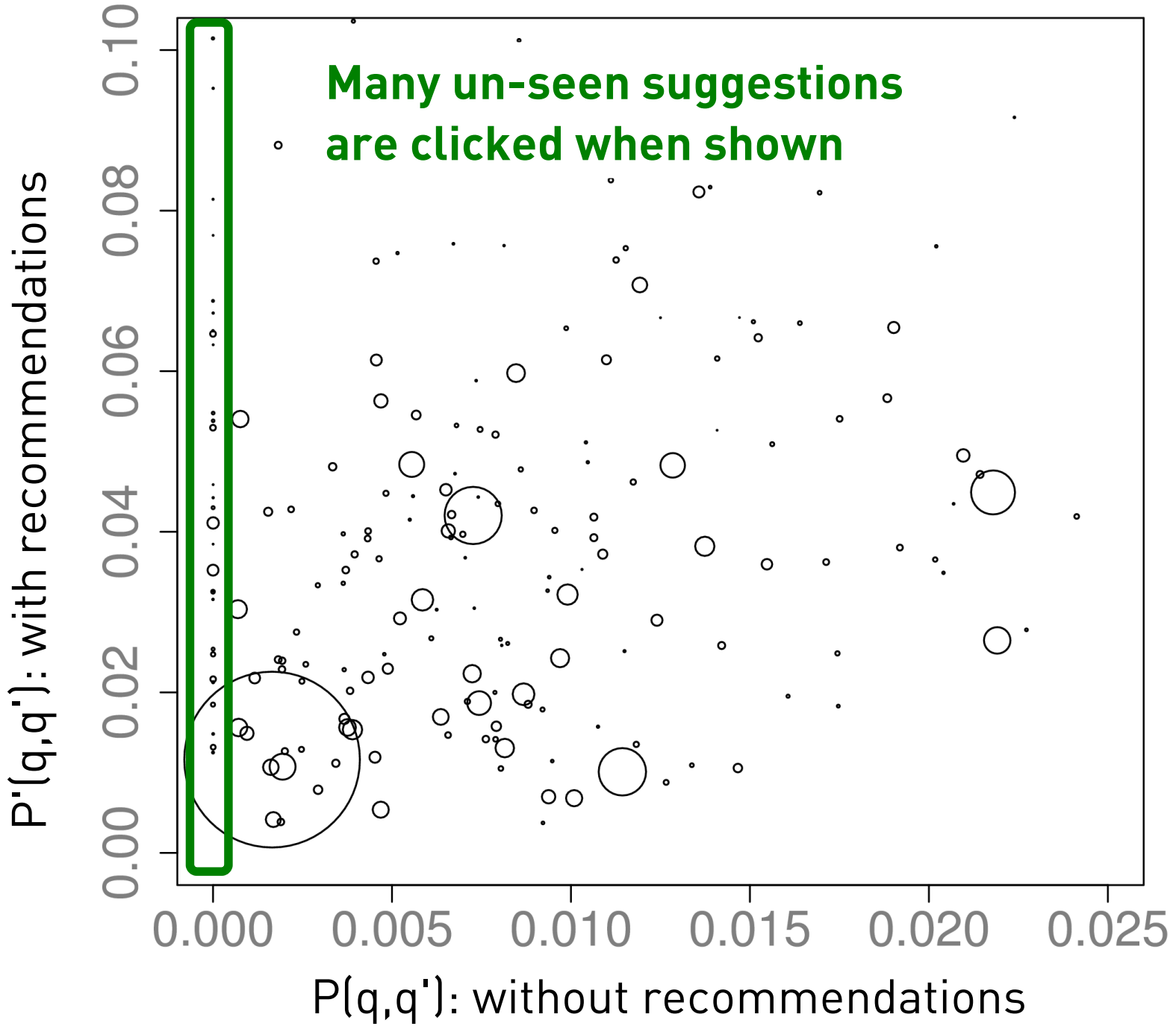

r= 0.95
0.4
0.5
0.6
0.7
0.8
0.9
τ'q
0.5
0.6
0.7
0.8
0.9
1.0τq−0.8∑ ρq,q'

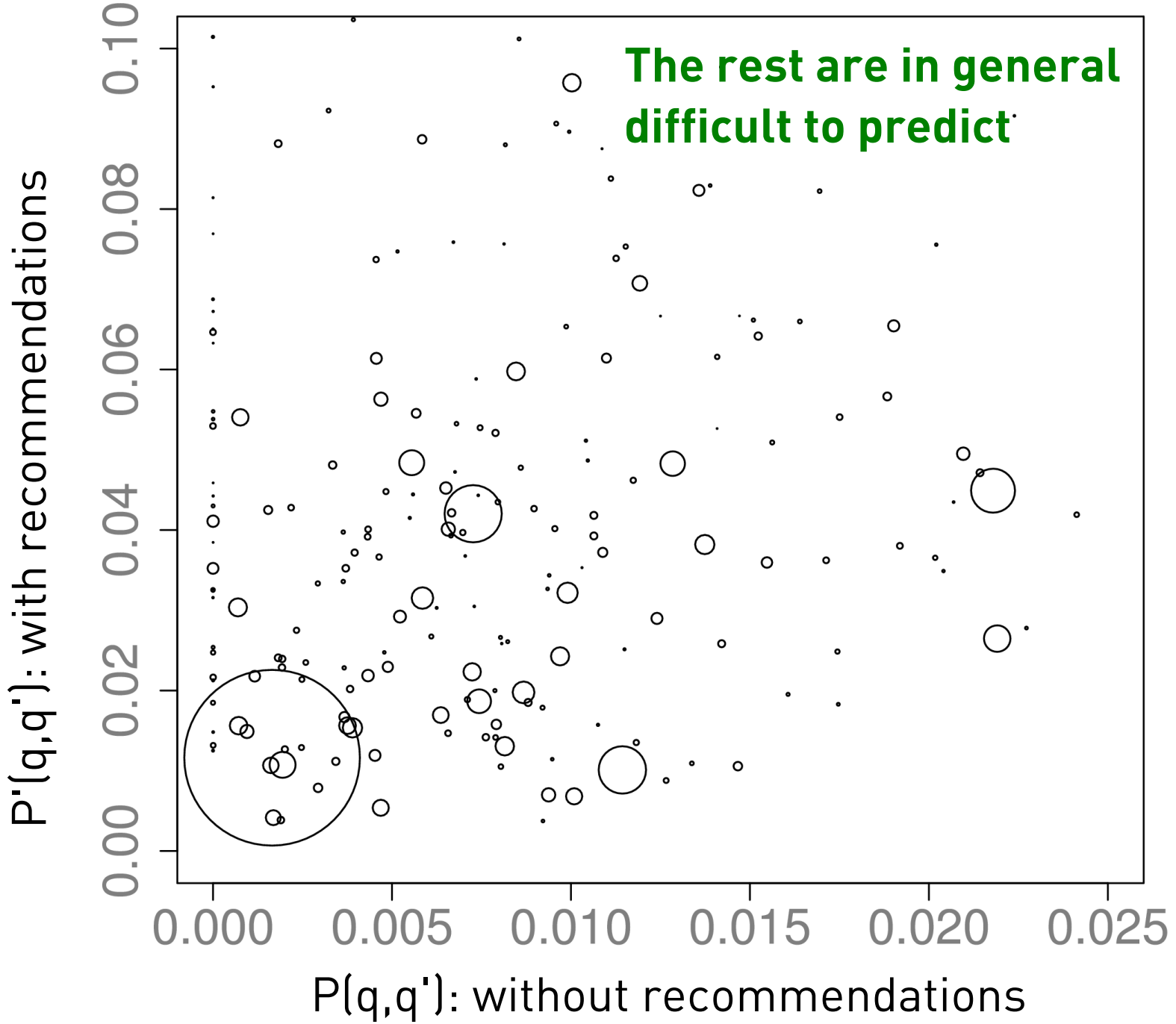
Empirical observations

- Recommendations decrease termination probability from ≈0.90 to ≈0.84
- Decrease is almost entirely due to more clicks on recommendations
- ρ is difficult to estimate


P'(q,q'): with recommendations
0.00
0.02
0.04
0.06
0.08
0.10
0.000
0.005
0.010
0.015
0.020
0.025
P(q,q'): without recommendations


P'(q,q'): with recommendations
0.00
0.02
0.04
0.06
0.08
0.10
0.000
0.005
0.010
0.015
0.020
0.025
P(q,q'): without recommendations
There is an increase


Many un-seen suggestions
are clicked when shown
P'(q,q'): with recommendations
0.00
0.02
0.04
0.06
0.08
0.10
0.000
0.005
0.010
0.015
0.020
0.025
P(q,q'): without recommendations


The rest are in general difficult to predict
P'(q,q'): with recommendations
P(q,q'): without recommendations
0.00
0.02
0.04
0.06
0.08
0.10
0.000
0.005
0.010
0.015
0.020
0.025

So what do we do?

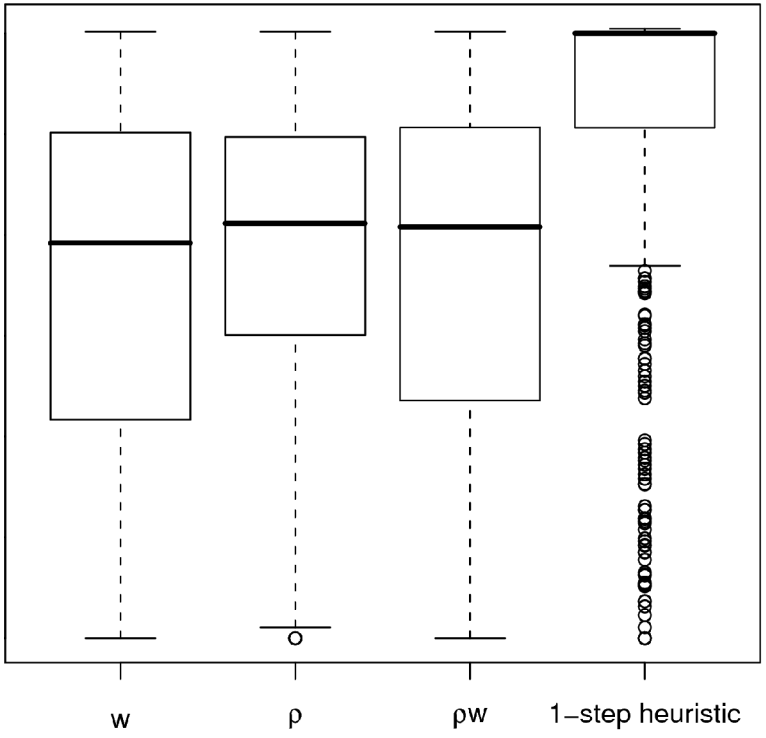
- We approximate ρ by a linear function on
  - P(q,q')
  - Textual similarity of q and q'
  - Terminal probability of q
- We have a low accuracy on this prediction
  - $r \approx 0.5$
- We use as weights the CTR on results

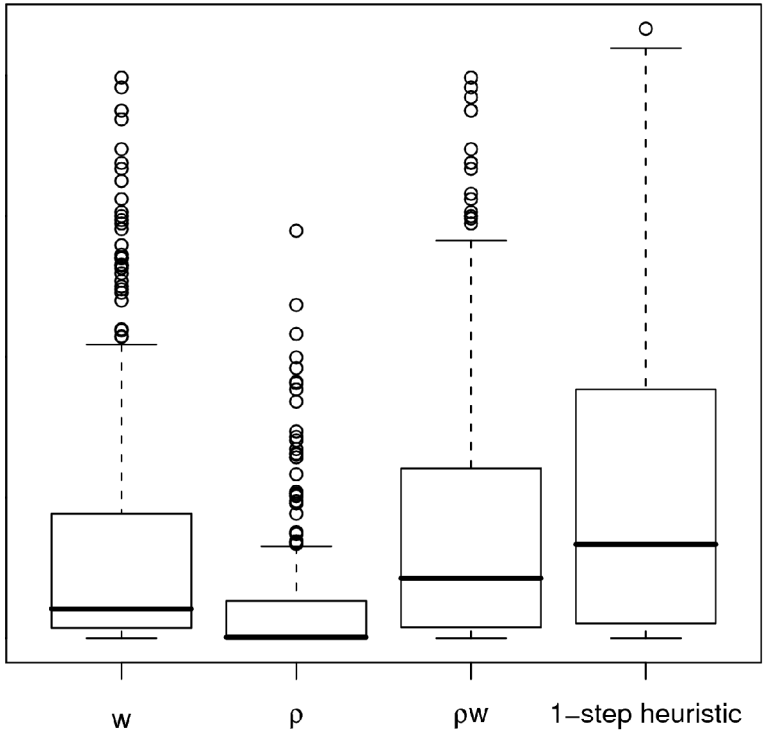
Evaluation

Evaluation results

- Baselines:
  - Prefer queries with large w
  - Prefer queries with large ρ
  - Prefer queries with large ρw
- Greedy heuristic performs better for both utility functions

Expected utility





"*Kavafian*" objective

"*Machiavellian*" objective

Evaluation results

- Greedy heuristic performs well
- What about relevance?
  - 420 queries assessed by 3 judges
  - There were no significant changes in relevance between systems

Conclusions

- General framework for "nudging" users in a certain direction
- Open algorithmic and practical questions
- In the paper: related work

An Optimization Framework for Query Recommendation*

Aris Anagnostopoulos[1] aris@cs.brown.edu
Luca Becchetti[1] becchett@dis.uniroma1.it
Carlos Castillo[2] chato@yahoo-inc.com
Aristides Gionis[2] gionis@yahoo-inc.com
[1]Sapienza University of Rome, Italy
[2]Yahoo! Research Barcelona, Spain

ABSTRACT

Query recommendation is an integral part of modern search engines. The goal of query recommendation is to facilitate users while searching for information. Query recommendation also allows users to explore concepts related to their information needs.

In this paper, we present a formal treatment of the problem of query recommendation. In our framework we model the querying behavior of users by a probabilistic reformulation graph, or query-flow graph [Boldi et al. CIKM 2008]. A sequence of queries submitted by a user can be seen as a path on this graph. Assigning score values to queries allows us to define suitable utility functions and to consider the expected utility achieved by a reformulation path on the query-flow graph. Providing recommendations can be seen as adding shortcuts in the query-flow graph that "nudge" the reformulation paths of users, in such a way that users are more likely to follow paths with larger expected utility.

We discuss in detail the most important questions that arise in the proposed framework. In particular, we provide examples of meaningful utility functions to optimize, we discuss how to estimate the effect of recommendations on the reformulation probabilities, we address the complexity of the optimization problems that we consider, we suggest efficient algorithmic solutions, and we validate our models and algorithms with extensive experimentation. Our techniques can be applied to other scenarios where user behavior can be modeled as a Markov process.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval—*Query formulation ; Search process*

*Partially supported by EU Project N. 215270 FRONTS and by MIUR FIRB project N. RBIN04TMB9: "Tecnologia e Scienza per le reti di prossima generazione".

WSDM'10, February 4–6, 2010, New York City, New York, USA.

General Terms

Algorithms

Keywords

Query reformulations; Query suggestions

1. INTRODUCTION

Query recommendations are a prominent feature of modern search engines. Query recommendations serve several purposes: correcting possible spelling mistakes, guiding users through their information-seeking tasks, allowing them to locate information more easily, and helping them explore other concepts related to what they are looking for.

The simplest form of query recommendation is *spelling correction*, a topic that we do not address in this paper. Instead we focus on more elaborate forms of query recommendations. For instance, by submitting the query "chocolate cookie" a user may be prompted to other queries such as "chocolate cookie recipe", or "chocolate chip cookie recipe", but also to related concepts such as "brownies", "baking", and so on.

A key technology for enabling query recommendations is query-log mining, which is used to leverage information about how people use search engines, and how they rephrase their queries when they are looking for information. Most of the proposed query-recommendation algorithms in the literature use aggregate user information mined from query logs and allowing to identify queries that are relevant to what the user is searching [2–4, 13, 14]. Current state-of-the-art methods often produce relevant query recommendations, but typically there is no clear objective to optimize and query-recommendation methods are fairly ad-hoc.

In this paper we propose a general and principled methodology for generating query recommendations. We model the query-recommendation problem as a problem of optimizing a global utility function. Our approach consists of the following ingredients:

- First, we assume that it is possible to aggregate historical information from a query log to build a query-reformulation graph [3]. The nodes of this graph are distinct queries, and an edge (q, q') is annotated with the probability that a user will submit query q' after submitting query q. We then model the querying behavior of users as random walks on this graph.
- Second, we assume that the queries in the query-flow graph have intrinsic score values $u(q)$, which model a

Q&A

chato@yahoo-inc.com