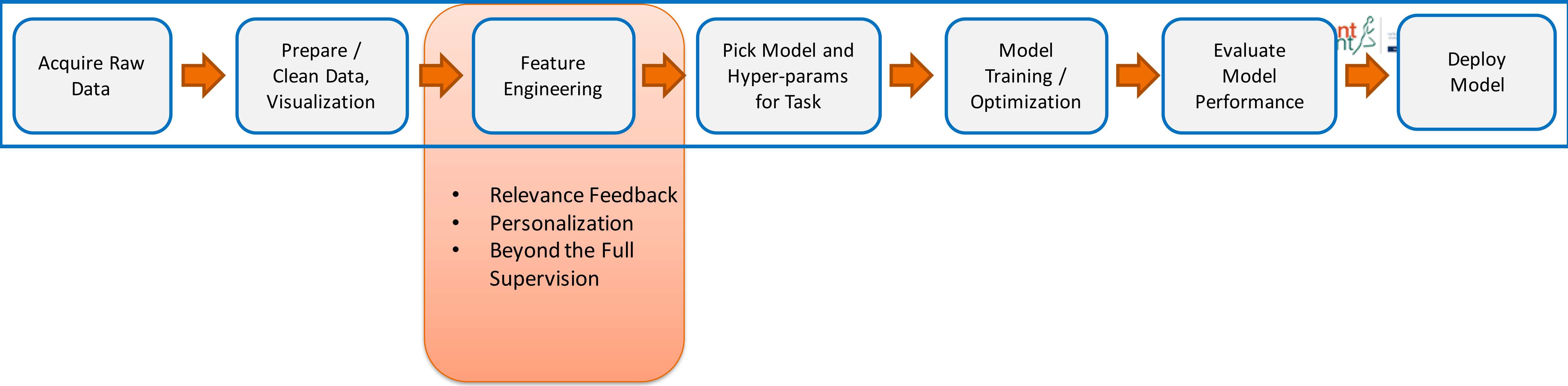
Focus for this lecture

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- Relevance Feedback
- Personalization
- Beyond the Full Supervision

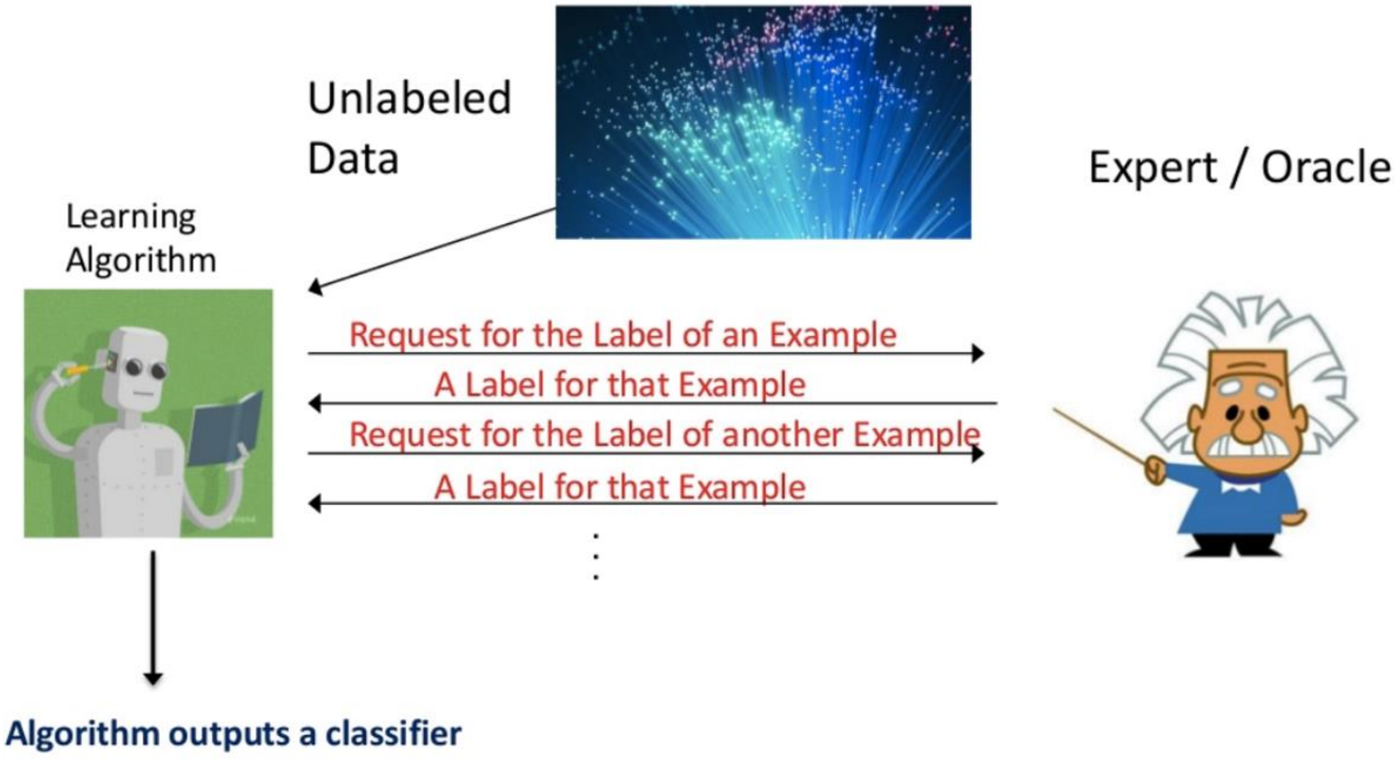


ML Systems with User in the Loop

Different Roles for Human in the Loop

- An expert in the loop during training a system
  - Active Learning
- An expert/human in the loop while deploying the system
  - Simpler tasks are first given to the machine.
- A human co-worker who shares the same task or complement with an ML system. Learns by watching human/expert
  - Incremental Learning, Intelligent Task Division
- A user who interacts with the system
  - Cooperative users, User Feedback; Personalization, incentivize

Learn with minimal # of examples ? Active Learning



Active Learning

- Stream-Based Active Learning
  - Consider one unlabeled example at a time Decide whether to query its label or ignore it
- Pool-Based Active Learning
  - Given: a large unlabeled pool of examples Rank examples in order of informativeness
  - Query the labels for the most informative example(s)

Q: HOW MANY SAMPLES ARE REQUIRED TO LEARN THE CONCEPT OF RECTANGLES?

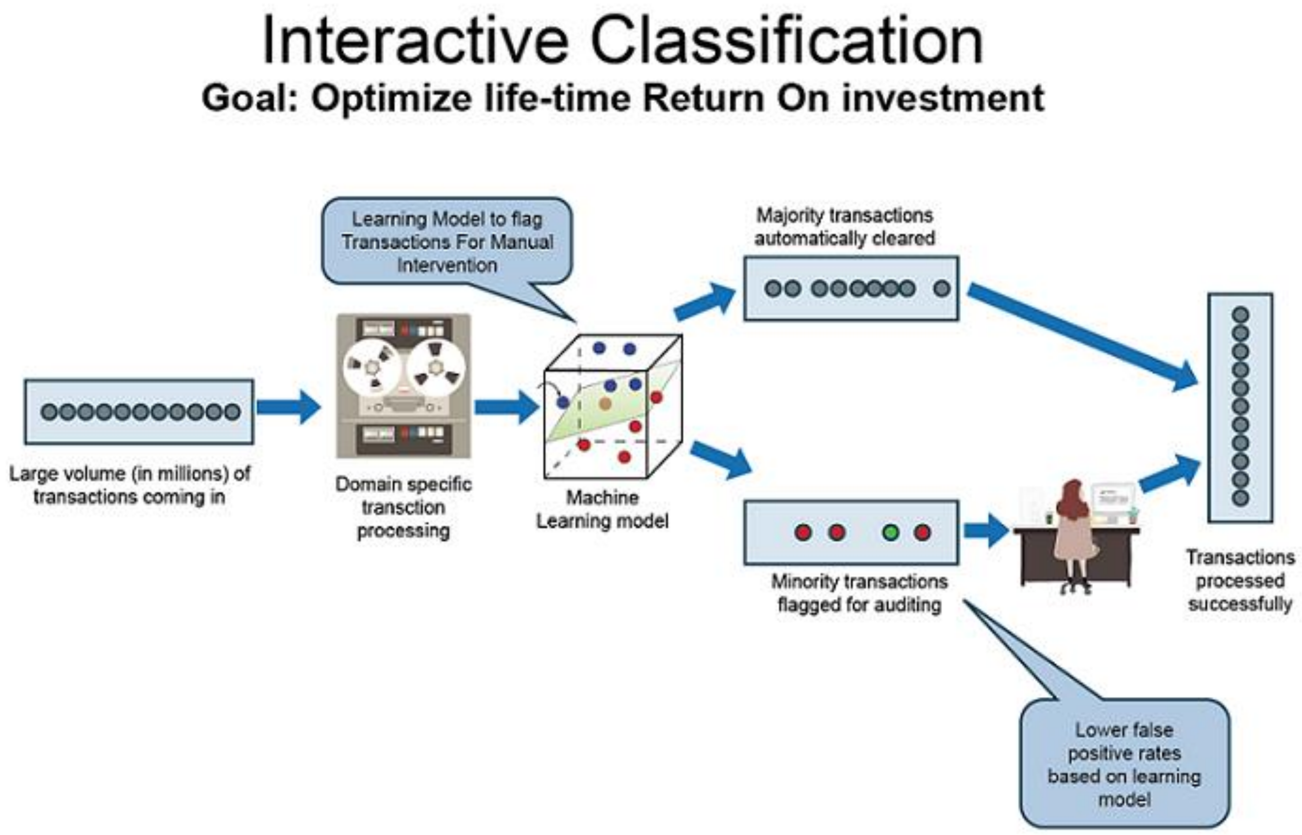
HOW DO I DEPLOY WHEN THE SYSTEM IS NOT 100% ACCURATE?

Situation

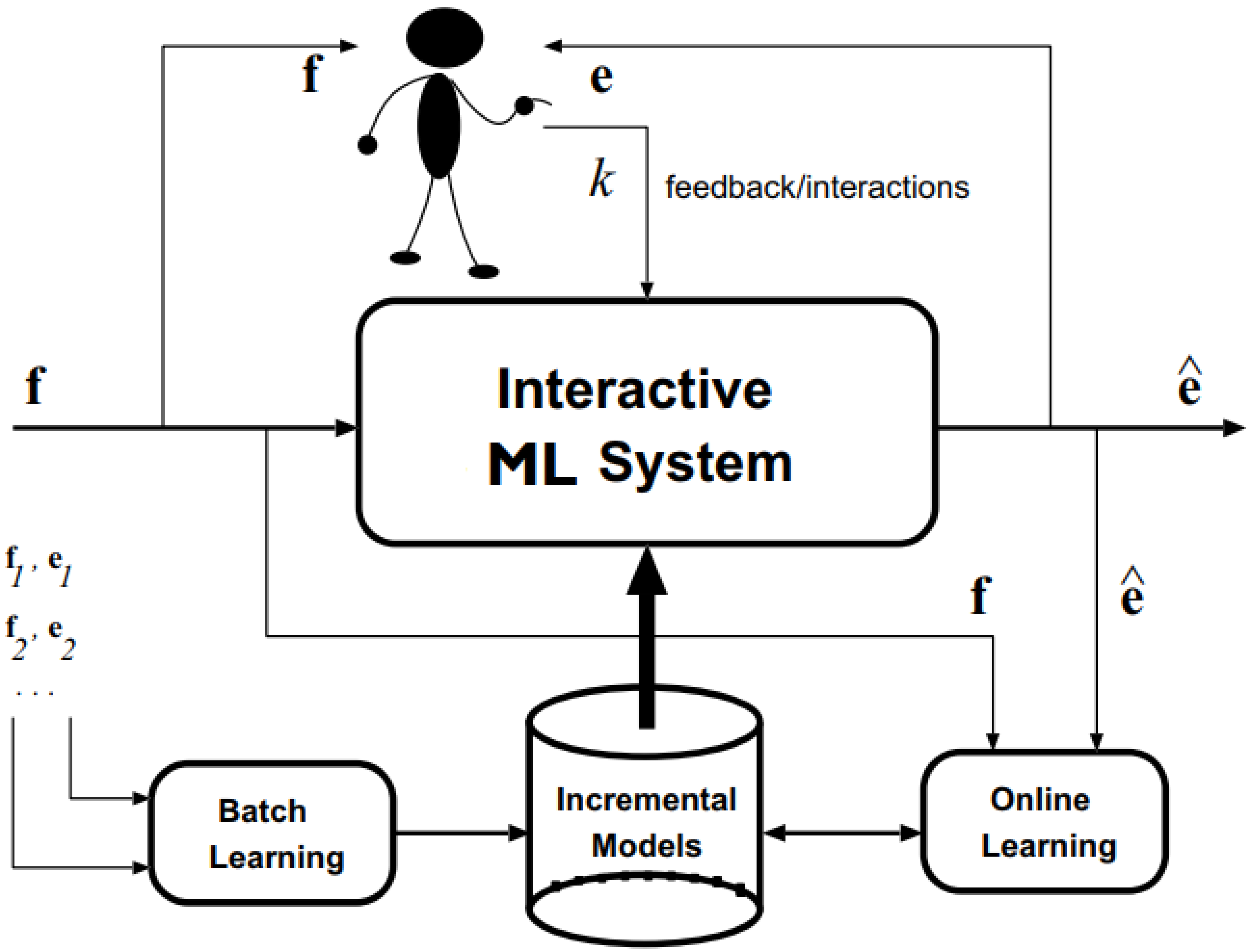
- System does easy tasks
  - System decides what is easy and what is difficult
- Harder tasks are given to humans
  - Human load reduces significantly
- With time
  - Specific hard situations are found and examples gets created
- System improves;
  - Automatically
  - With expert interventions

Users in the ML systems

- Defining Characteristics
  - Expensive domain experts
  - Skewed class distribution(minority events)
  - Concept/ Feature drift
  - Biased sampling of labeled historical data
  - Lots of unlabeled data



Another Scenario (Interactive ML)



User may adjust:
- K (control) or
- Directly give desired e

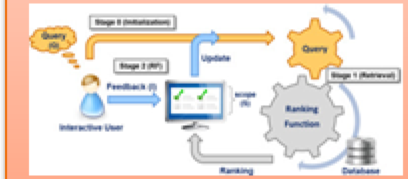
Associated Issues

- Incremental and Computational Issues
  - How do we learn, adapt and forget
  - What is the basic knowledge and what do we adapt?
- Stability
  - Am I overlearning and changing too fast?
  - Stability, convergence and other algorithmic issues.

Q: THREE EXAMPLES WHERE SYSTEM CAN "SLOWLY" TAKE OVER FROM HUMANS

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- Relevance Feedback

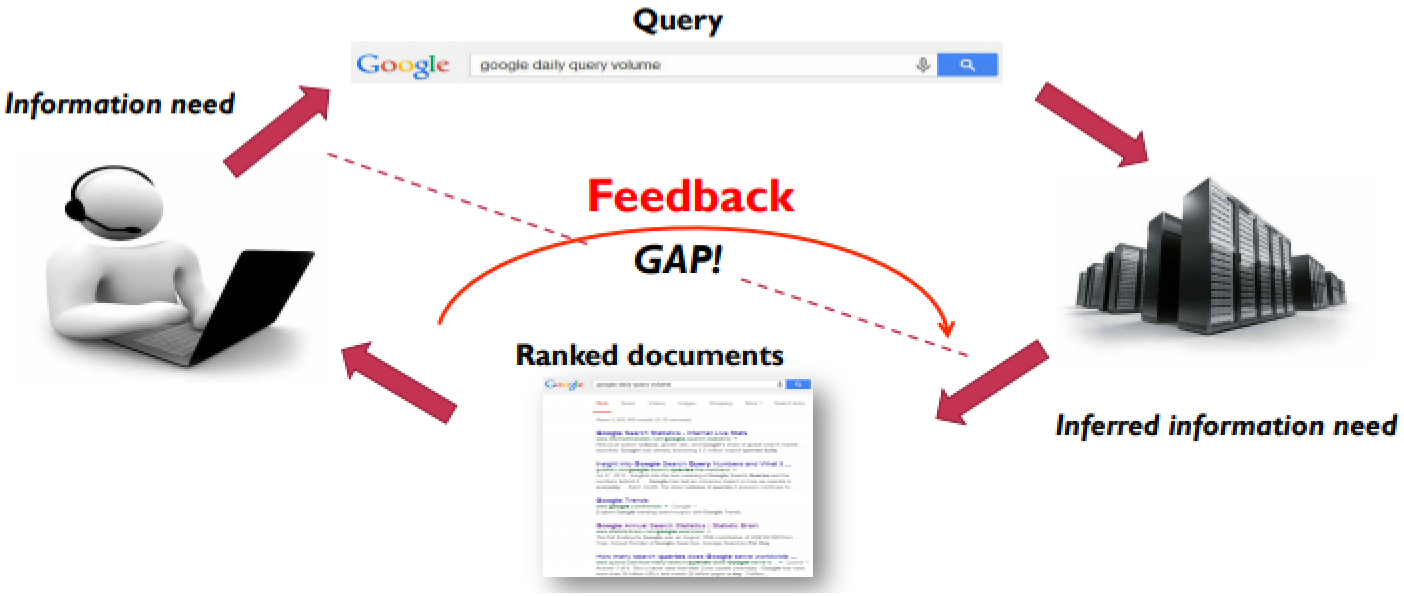


Relevance Feedback

(Interactive Classification)

User Feedback

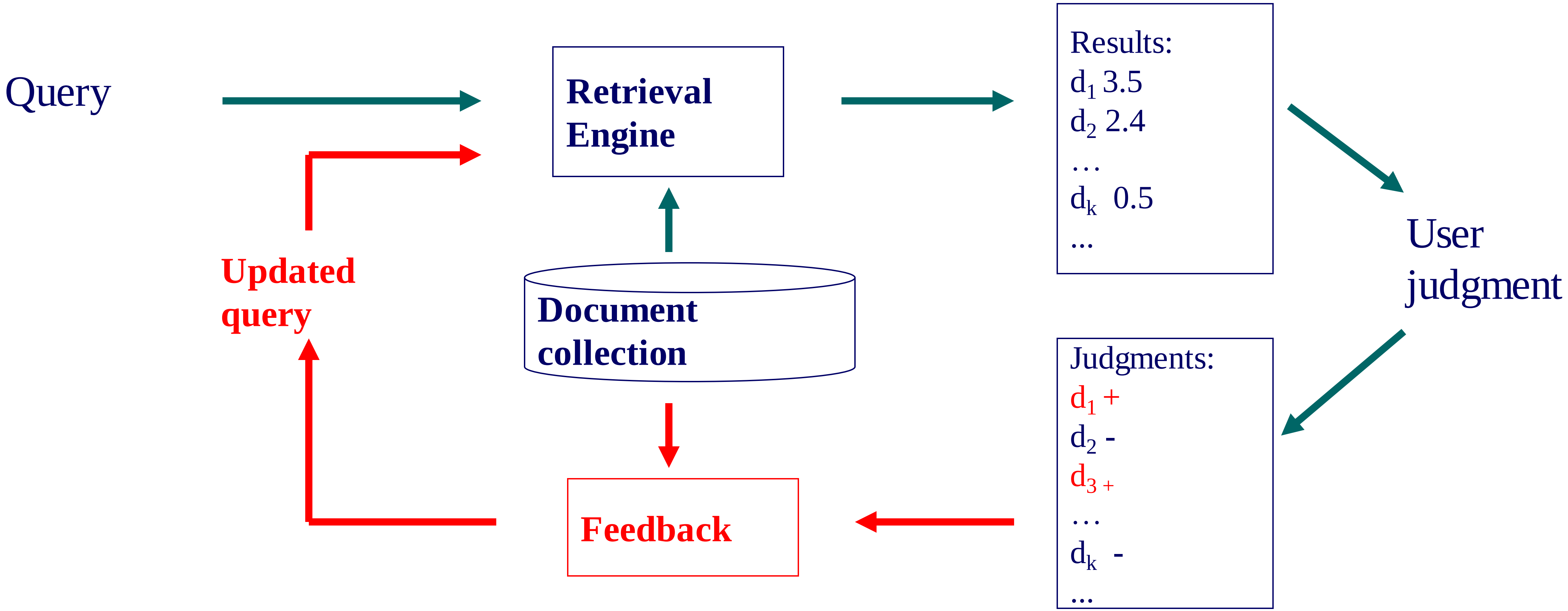
- An IR system could be an interactive system



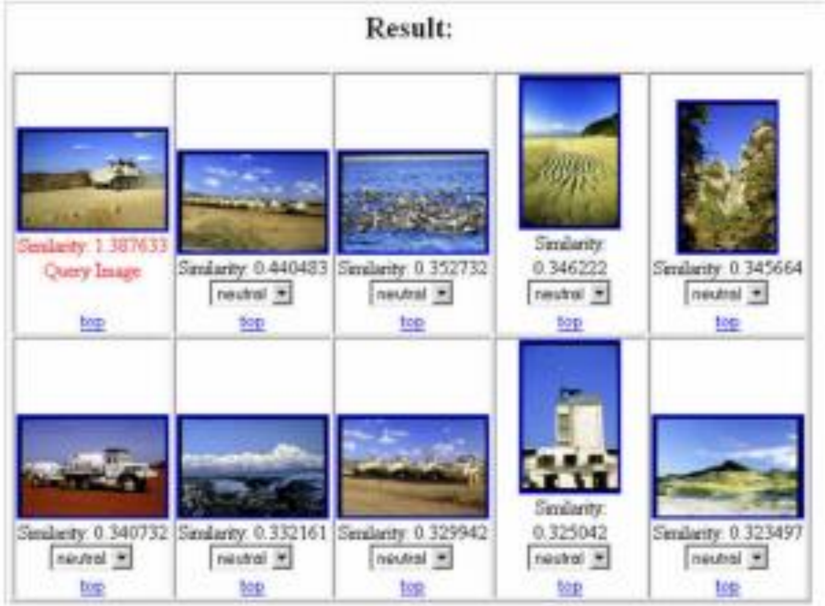
Use Scenario

- A query q or a classifier w is given.
- Search engine retrieves a set of possible answers.
  - x1, x2, x3, etc.
- System guess the user intend and improve the answers.
  - x7,x12,x23, etc.
- User is able to smartly navigate and get what she is looking for.
- E.g. Search for a specific fashion/design in a large database.

Relevance feedback



Effective and Popular(?)

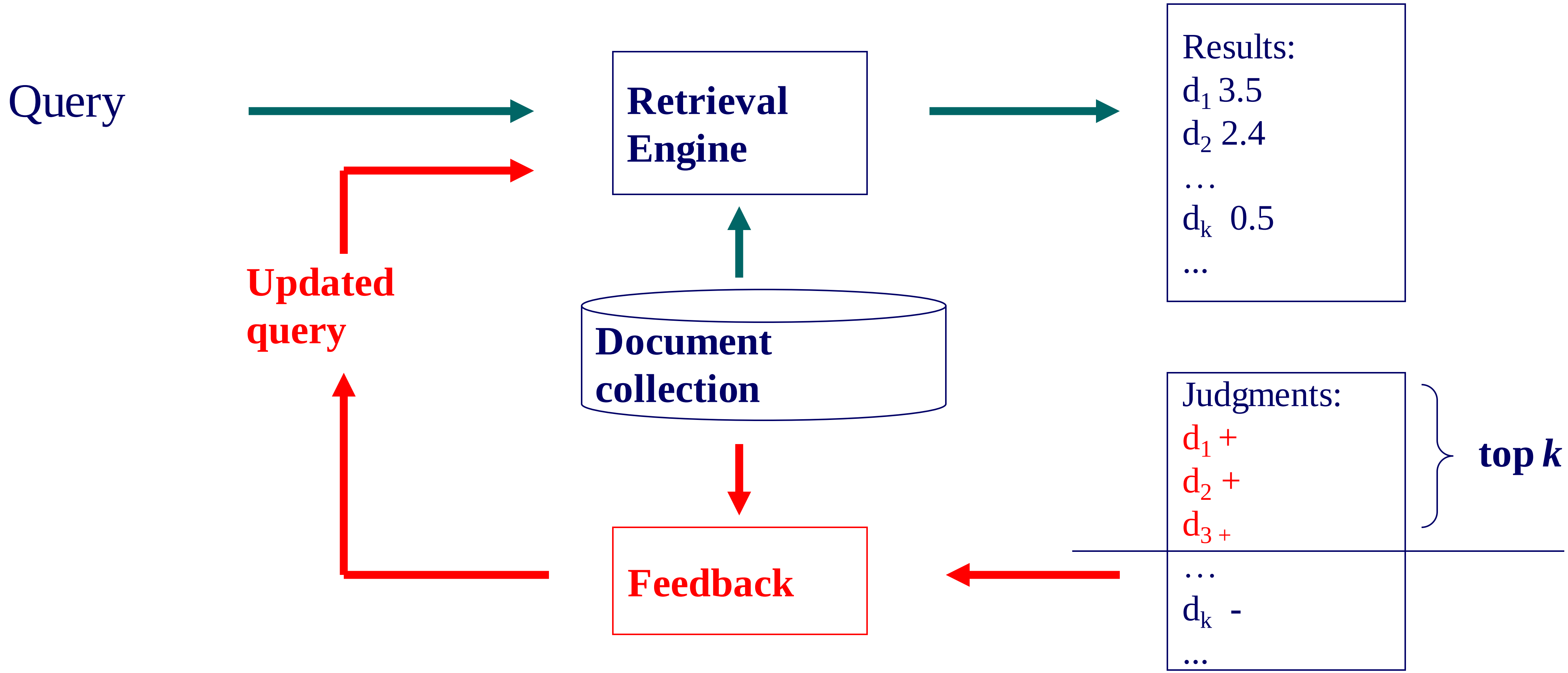


Too Explicit?

Code Available

Pseudo feedback and Query Expansion

- What if the users are reluctant to provide any feedback



Rocchio Model

$$Q_1 = \alpha \; Q_0 + \frac{\beta}{n_1}\sum_{i=1}^{n_1} R_i - \frac{\gamma}{n_2}\sum_{i=1}^{n_2} \; S_i$$

where

Q_0 = the vector for the initial query

R_i = the vector for the relevant document i

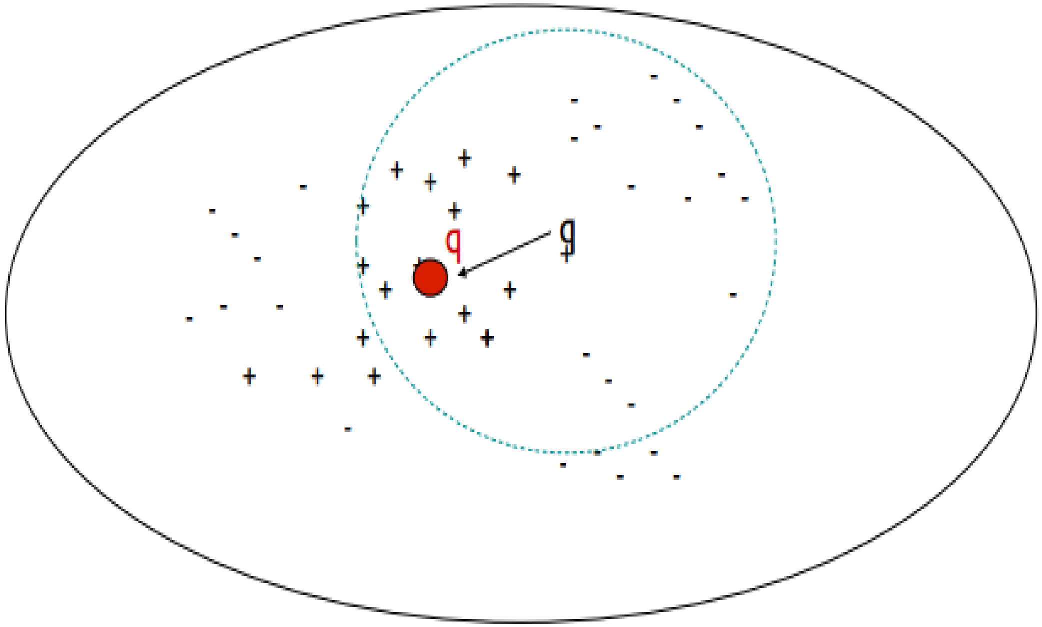
S_i = the vector for the non-relevant document i

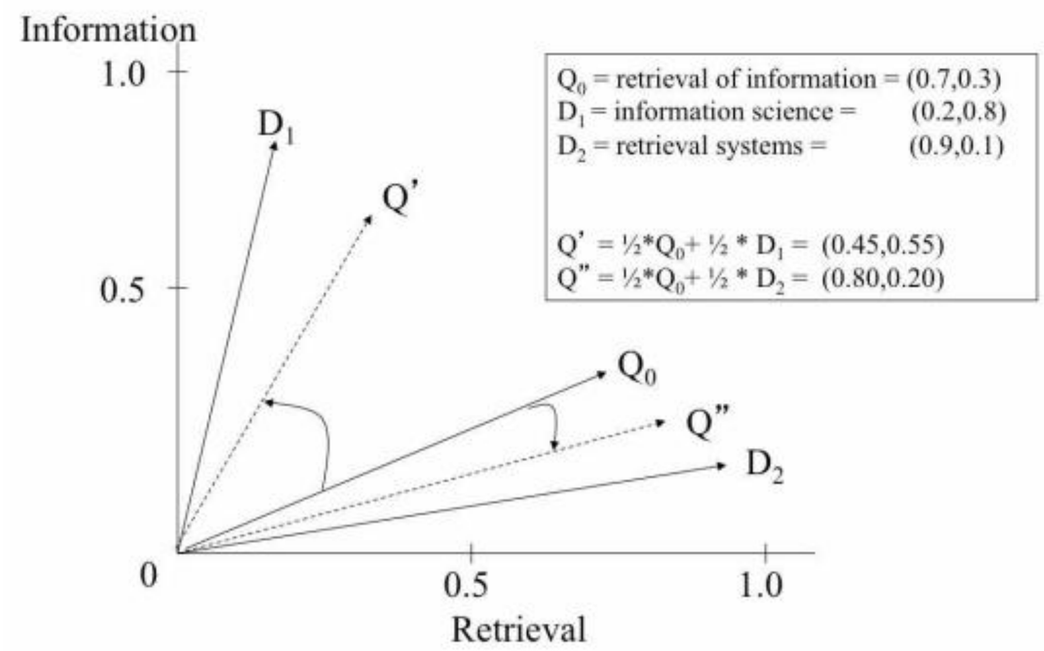
n_1 = the number of relevant documents chosen

n_2 = the number of non-relevant documents chosen

α, β and γ tune the importance of relevant and nonrelevant terms (in some studies best to set β to 0.75 and γ to 0.25)

Illustration





Challenges and Refinements

- Can we force user to say + and – on the answers?
  - Often + is more clear ?. But not –ve is not shared.
  - Cases when only + or Only – is available.
- Often + is implicit (I click/browse) and not explicit.
- Examples:
  - Browsing for fashion (clothes)

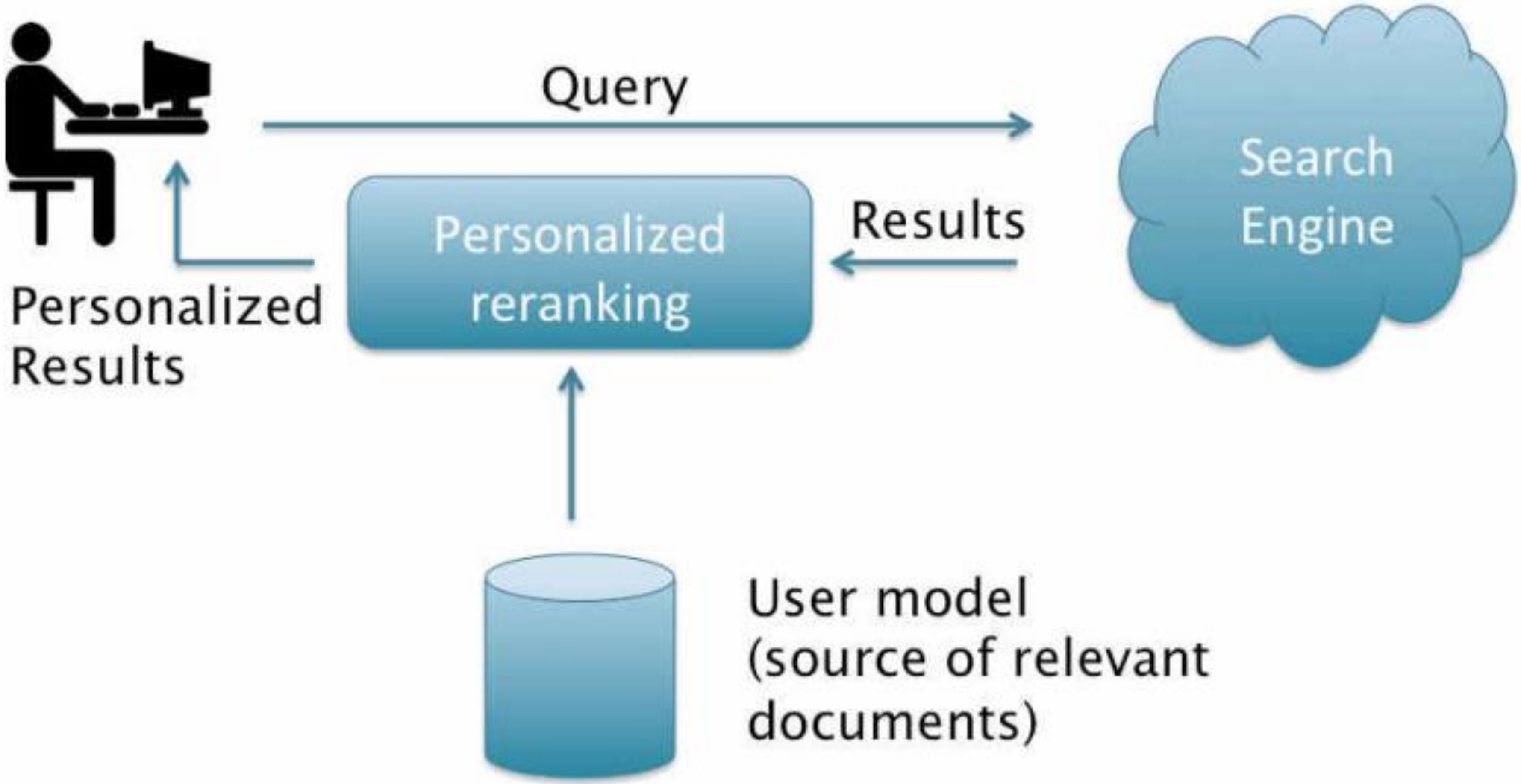
Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

✓ Relevance Feedback
- Personalization

Personalization


DON'T WORRY, IT'S ONLY MARKETERS COLLECTING OUR PERSONAL DATA SO THEY CAN CREATE MORE RELEVANT ADVERTISING FOR US.
TOM FISHBURNE

Use Case (RF style)



Many Relevant Attributes

- Consider the following pieces of information
  - Geographical Location
  - Age, gender, ethnicity, religion, etc.
  - Interests
  - Previous reviews on products
  - ……
- How could these pieces of information help?
- How to collect these information?

Approaches

- Individual Vs Collaborative
- Reactive Vs Proactive
- User Vs Item Information

Individual Vs Collaborative

- Individual approach (E.g. Google Personalized Search)
  - Use only individual user's data
  - Generate user profile by analyzing
    - User's browsing behavior
    - User's active feedback on the system
- Advantage
  - Can be implemented on the client-side - no privacy violation
- Disadvantage
  - Based only on past interactions.

Reactive Vs Proactive

- Reactive approach
  - Explicitly ask user for preferences
    - Either in the form of query or feedback
- Proactive approach
  - Learn user preferences by user behavior
    - No explicit preference demand from the user
- Behavior is extracted
  - Click-through rates
  - Navigational pattern

User Vs Item Information

- User Information
  - Geographic location (from IP address)
  - age, gender, marital status, etc. (explicit query)
  - Lifestyle, etc. (inference from past behavior)
- Item Information
  - Content of Topics – movie genre, etc.
  - Product/ domain ontology

Summary

- ML systems can work with human
  - While Training
  - While Testing
  - To Learn, To Help
- A strategy to deploy and slowly improve
- A strategy to co-exists with humans
  - From Indian view point
  - From global view point

Questions?

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

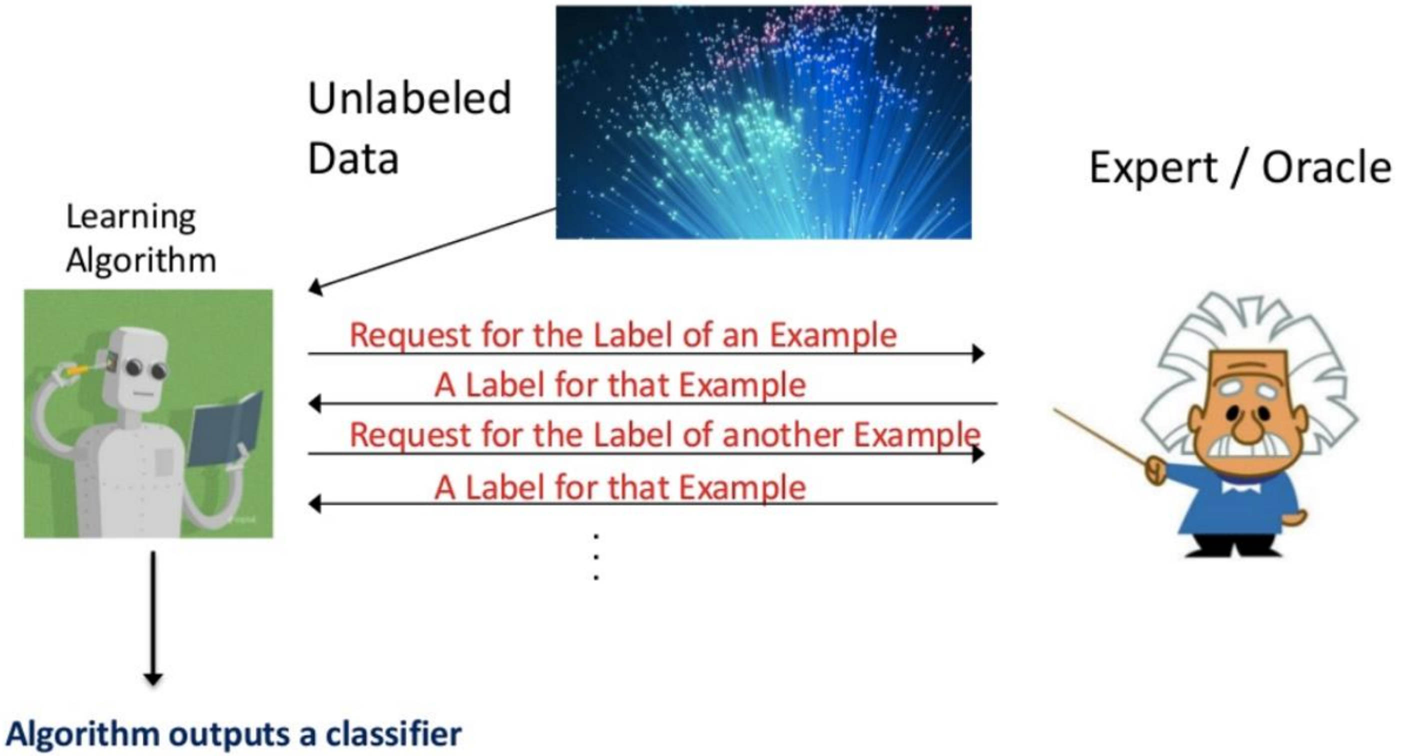
- ✓ Relevance Feedback
- ✓ Personalization
- • Beyond the Full Supervision

Beyond the Full Supervision

Challenges with Supervision

- I have too much data. But most of them are unlabeled. What do we do?
- I have labeled data. But a good percentage of the labels are erroneous. What do I do?
- I have libeling's from experts itself. But they do not agree. What do we do?
- My supervisors are too costly. How do I do minimize the cost of supervision?
- …

Learn with minimal # of examples ?



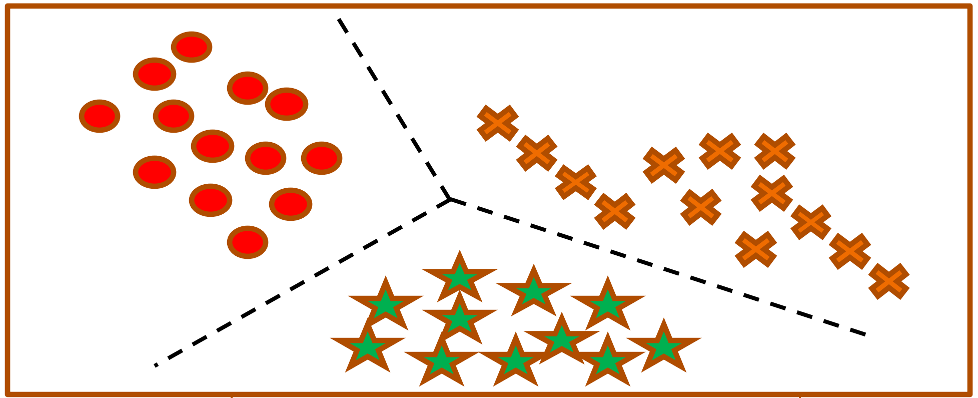
Active Learning

- Eg. Learn the notion of a rectangle.

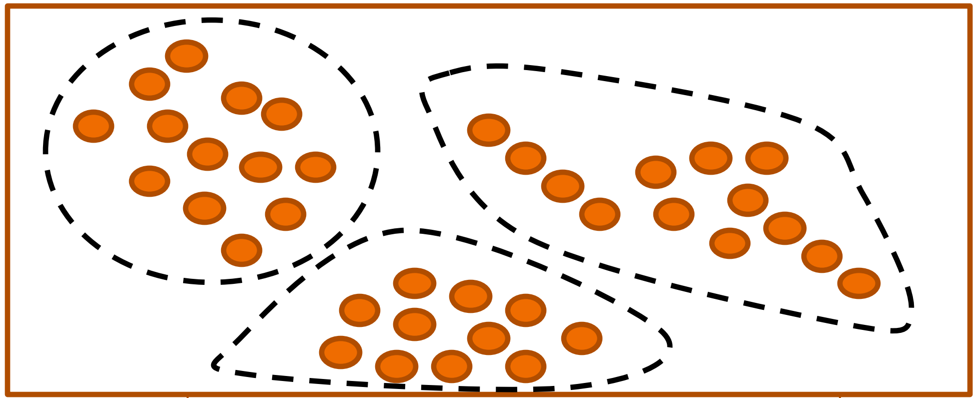
Semi Supervised Learning

- I have a small quantity of labelled data and large quantity of unlabeled data.
  - How do I take advantage of the unlabeled data?

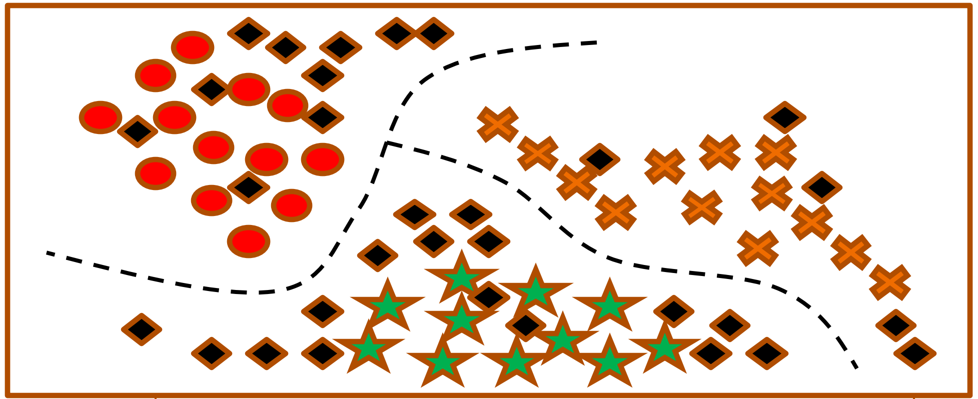
Algorithms



Supervised learning



Unsupervised learning



Semi-supervised learning

Self Training: Naïve

- Train a supervised learner on available labelled data (X_l, Y_l).
- Label all points in unlabeled data X_u.
- Retrain the classifier using the new labels for documents where the classifier is most confident.
- Continue until labels do not change any more.

Self Training: Refined

- **Assumption:** One’s own high confidence predictions are correct.
- **Self-Training Algorithm**
  - Train on labeled examples
  - Predict on unlabeled examples
  - Add (x, f(x)) to the labeled data
    - Add all
    - Add a few most confident pairs
    - Add weight for each pairs
  - Repeat the process

Co-Training

- Co-training assumed two "Views" of the data where each input x is a pair

$$x = (x_1, x_2)$$

- Eg. In the context of web page classification,
  - x_1 may be metadata associated with the web page such as title etc.
  - x_2 be the words in the link pointing to this page.
- Assume there exists functions c_1, c_2 and c such that

$$c_1(x_1) = c_2(x_2) = c(x)$$

- Two sets of features x_1 and x_2 are conditionally independent given the class.

1998 paper demonstrates, with 12 labeled examples,788 web pages could be classified with 95% accuracy.

Co-Training

- Use the labeled data to learn the initial h_1, h_2
- First use h_1 to label examples that it is confident about and then feed these to our learner to update h_2
- Then use h_2 to label examples that it is confident about and then feed these to our learner to update h_1
- Keep repeating this process

Summary: Questions?

- Varying amount and quality of supervision
  - Many wrapper style methods.
  - Intuitive
- Many principled formulations
  - Formal extensions of existing methods
    - (eg. Transductive SVMs; Semi Supervised Random Forest)
  - Many newer learning problems
    - (eg. Multiple Instance Learning,)

Summary

Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- ✓ Relevance Feedback
- ✓ Personalization
- ✓ Beyond the full Supervision

Thanks!!

Questions?