Designing for End-User Interactive Machine Learning

Saleema Amershi, University of Washington
James Fogarty, University of Washington
Ashish Kapoor, Microsoft Research
Desney Tan, Microsoft Research
Facing Large Unstructured Data Sets

People are increasingly faced with a need to interact with large collections of unstructured data

The Web

Data collected from sensing equipped devices

Massive scientific datasets

Large personal stores (e.g., documents and images)
Dealing with Large Unstructured Data Sets

Developers have use machine learning to:

• Detect *spam* in web pages.
• Detect *walking, running, and biking* in sensory data.
• Detect *faces* and *cars* in images.

But, what if I want to:

• Detect *well-designed web pages*?
• Detect when my advisor is *approaching my office*?
• Detect images of *puppies* playing in the *park*?
Dealing with Large Unstructured Data Sets

The challenge is not any particular concept, but that developers cannot possibly foresee the countless variety of distinctions people might want to make.

One potential solution:

Enable end-users to define concepts themselves, via end-user interactive machine learning.
End-User Interactive Machine Learning

- Data
- Classifier
- Person provides examples of concept
- Machine classifies remaining data
We know *we can create* systems that enable end-user interactive machine learning

- Selection Classifier [Ritter & Basu, 2009]
- Crayons [Fails & Olsen, 2003]
- A CAPPella [Dey et al., 2004]
- Exemplar [Hartmann et al., 2007]

But, an open question is *how should we design* the actual *interaction* with end-user interactive machine learning.
Designing for End-User IML in CueFlik

- Web Images
  - System re-ranks images based on examples
  - Person provides positive and negative examples
- Classifier
Looking for Product Photos...

- stereos
- bikes
- drinks
~10% out of first 200 are matches
~12% out of first 200 are matches
CueFlik
How should we design end-user interactive machine learning in CueFlik?
How should we illustrate current model?

**Single Presentation**

User labeled positive examples

User labeled negative examples

All images ranked
How should we illustrate current model?

**Split Presentation**

User labeled positive examples

50 best matches

User labeled negative examples

50 worst matches
Evaluations

Given 1000 images previously retrieved for a keyword query (e.g., “stereo”) Train CueFlik to recognize a concept (e.g., “product photo”)

Images with Products on a White Background

Sheet of paper with concept description and target images given to participants

Initial distribution of target images
Measuring Model Quality

Example initial positions of target images

Classifier trained by participant with CueFlik

Example final positions of target images
Presentation Style Matters

**Average Task Time**

- Split: 120 seconds
- Single: 140 seconds

F(1, 11)=8.77, p ≈.013

**Average Model Quality**

- Split: 250 Improvement in Rank
- Single: 150 Improvement in Rank

F(1, 11)=6.90, p ≈.024
We hypothesized *Split* presentation was valuable because it exposed a manageable sample of positive and negative matches. However, best/worst matches poorly illustrate positive and negative classes give the system little additional information.
Global Overviews

Global overviews try to show representative examples, maximizing mutual information between selected and unselected instances.
Projected overviews illustrate the principal axes of variation in a region, sampling representative examples along each principal dimension.
Sample Overviews

High Certainty
Sample Overviews

Global
Sample Overviews

Projected
Example Selection Matters

**Average Task Time**

- High Certainty: F(1, 170) = 7.72, p ≈ 0.006
- Overviews: F(1, 170) = 8.86, p ≈ 0.003

**Average Model Quality**

- High Certainty: F(1, 170) = 7.72, p ≈ 0.006
- Overviews: F(1, 170) = 8.86, p ≈ 0.003
Not a Simple Time Accuracy Tradeoff

![Graph showing improvement over time for High Certainty and Overviews.]
Model *Decay from Best to Final*

Overviews led to better *best* models in the same amount of time

\[ F(1, 170)=6.52, \ p \approx .012 \]

Overviews reduce the magnitude of the decay from *best* to *final*

\[ F(1, 170)=3.44, \ p \approx .065 \]
Overview Results

Participants noted this period of *Decay*

“it was weird, sometimes it would start out doing really well, but as I kept going it did worse”

“it is hard to know if more data is better as I should probably stop occasionally to see the results as I am going”
How can we prevent decay?

Research on interactive machine learning has implicitly assumed a simple feedback loop.

Instead of a focus on providing additional examples to improve the current model, what if interaction was about choosing from multiple potential models?
History and Revision in CueFlik

Overview

History of Model Versions
Plotting LOOCV and Top Images for Each Version

Overview

Undo/Redo

Full Ranking
History Results

History plot was problematic for end-users

“I felt like I was concentrating too much on the line graph”

“The graph was sometimes really random ... Even when I thought the [ranked results] were better, the program itself did not think so”

But using the snapshots to go back was helpful

“[the history] helped because when I started to get off track I could always go back and try a different route”

Did not improve the quality of resulting models
Revision Matters

End-users readily adopted snapshots and undo/redo

Used Revision features in 41% to 68% of trials

End-users considered Revision a natural component

“[without revision] it felt a little like typing on a keyboard without a backspace key”

Conditions with Revision led to better Final scores

F(1, 119)=3.57, p ≈.061
Summary

Interactive concept learning is a powerful general strategy with many potential end-user applications.

Requires addressing both machine learning and human-computer interaction aspects of problems.

More details available in UIST 2009 paper

“Overview-Based Example Selection in End-User Interactive Concept Learning”
Thank You

Saleema Amershi, James Fogarty
Computer Science & Engineering
DUB Group
University of Washington
{samershi, jfogarty}@cs.washington.edu

Desney Tan, Ashish Kapoor
Microsoft Research
{desney, akapoor}@microsoft.com

This work was supported in part by the National Science Foundation under grant IIS-0812590.