Learning from evolving streams

Online triage of bug reports

Grzegorz Chrupala

Spoken Language Systems
Saarland University

EACL 2012
Issue trackers

- Used to track bugs or feature requests in software projects
- May receive hundreds of reports per day
- Need to be *triaged*: labeled and assigned developers
- Domain-specific challenges
Issue 116952: No search engine set as default
16 people starred this issue and may be informed of changes.

Status: Started
Owner: iva...@chromium.org
Cc: avayvod@chromium.org, jhaw...@chromium.org, anna...@chromium.org, levin@chromium.org, mirandac@chromium.org, msw@chromium.org, jeff...@chromium.org, pawli...@chromium.org, pkasting@chromium.org, rohit...@chromium.org, she...@chromium.org, isherman@chromium.org, sky@chromium.org, mrossetti@chromium.org, gwil...@chromium.org, iva...@chromium.org, dhollowa@chromium.org, ali...@chromium.org, sk...@google.com, gideonw...@chromium.org

Reported by sumon.ar...@gmail.com, Mar 6, 2012

Chrome Version : 19.0.1055.1

URLs (if applicable):

What steps will reproduce the problem?
1. Browse some sites.
2. Then enter text in Omnibox related to those sites you've just browsed.

What is the expected result?
There should be an option to search the keyword in Google.

What happens instead?
Only history URL appeared. No search option.

UserAgentString: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.24 (KHTML, like Gecko) Chrome/19.0.1055.1 Safari/535.24
Automate

- Predict project **SUBCOMPONENT** labels
- Predict developers **ASSIGNED** to bugs
As social media

- **Issue trackers:**
  - very specialized social media

- **Practices (labeling, triage):**
  - Negotiated explicitly
  - Emerging via imitation
  - Influenced by automation
Concept drift

- Practices evolve
- Software projects mature
- People involved come and go

For a learner, input and output change over time.
Contributions

- Collect data from modern software projects
- Analyze concept drift
- Apply state-of-the-art online learning and improve on current approaches
## Data

- Alternate items assigned to dev and test
- Dev sets sizes:

<table>
<thead>
<tr>
<th>Tracker</th>
<th>Output</th>
<th># Items</th>
<th># Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromium</td>
<td>Subcomponent</td>
<td>31,953</td>
<td>75</td>
</tr>
<tr>
<td>Chromium</td>
<td>Assigned</td>
<td>16,154</td>
<td>591</td>
</tr>
<tr>
<td>Android</td>
<td>Subcomponent</td>
<td>888</td>
<td>12</td>
</tr>
<tr>
<td>Android</td>
<td>Assigned</td>
<td>718</td>
<td>72</td>
</tr>
<tr>
<td>Firefox</td>
<td>Assigned</td>
<td>12,733</td>
<td>503</td>
</tr>
<tr>
<td>Launchpad</td>
<td>Assigned</td>
<td>18,634</td>
<td>1,970</td>
</tr>
</tbody>
</table>
Evolving class distribution

Chromium Subcomponent
Evolving class distribution

Launchpad Assigned
Progressive validation

For $i = 1$ to $\infty$

- Send input $i$ to learner
- Receive prediction $i$ and record error $i$
- Send true output $i$ to learner

$\text{Error}(n) = \sum_{i=1}^{n} \text{error}(i)$
Evaluation of ranking

- **Triage assistant**
  - show user a ranked list of suggested targets
- **Mean reciprocal rank**

\[
\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \text{rank}(i)^{-1}
\]
Features

- **Title** unigram and bigram counts
- **Description** unigram and bigram counts
- **Author ID**
- **Year, month and day** of submission
Baselines

- **Window** frequency
  - Relative class frequencies in previous $k \in \{100, 1000\}$ items

- **SVM** minibatch
  - Retrain every $n = 100$ steps on previous $k = 1000$ items

- **Perceptron**
  - Single pass, constant learning rate
Bugzie

- Based on a fuzzy set membership function:

\[
\mu(y, X) = 1 - \prod_{x \in X} \left( 1 - \frac{n(y, x)}{n(y, \cdot) + n(\cdot, x) - n(y, x)} \right)
\]

- Counts \(n(\cdot, \cdot)\) updated incrementally
- Feature cache: keep track of \(k\) most significant features
Regression SGD

- SGD with square loss as basic learner.

\[ w^{(t+1)} = w^{(t)} - \eta(t) \nabla L(y^{(t)}, w^{(t)T} x^{(t)}) \]

\[ L(y, \hat{y}) = (y - \hat{y})^2 \]

- Adaptive, per-feature learning rate (Duchi et al. 2010, Streeter and McMahan 2010)

- Learning rate larger for infrequent features.
Reduction from multiclass

- One-versus-all reduction

\[ T(x, y) = \left\{ (x', I(y = y')) \mid y' \in Y, \ x'_{h(i, y')} = x_i \right\} \]

- \( h(i, y') \) composes the index \( i \) with the label \( y' \) by hashing.
Summary of results (test)

G. Chrupała (Saarland Uni)  Learning from streams  EACL 2012  18 / 23
Chromium Subcomponent
Data becomes more difficult around 9,000
Launchpad Assigned

- Little concept drift
- $\approx 2000$ labels: hashing collisions
Best improvement over Window

![Bar chart showing improvements over different categories](chart.png)

G. Chrupala (Saarland Uni)  Learning from streams  EACL 2012  22 / 23
To conclude

- **Concept drift** is a crucial concern
- Modern online learner successfully **tracks stream evolution**
- Data available at: www.lsv.uni-saarland.de/resources.htm
- Ready to go **beyond bag-of-words**