

Learning from evolving streams

Online triage of bug reports

Grzegorz Chrupała

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 notified of changes.

[Back to](#)

Status: **Started**

Owner: iva...@chromium.org

Cc: avayvod@chromium.org,
jhawk...@chromium.org,
anna...@chromium.org,
levin@chromium.org,
mirandac@chromium.org,
msw@chromium.org,
jeffr...@chromium.org,
pawli...@chromium.org,
pkasting@chromium.org,
rohit...@chromium.org,
shess@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 simon.arn...@gmail.com, Mar 6, 2012

Chrome Version : 19.0.1055.1
OS Version: 6.1 (Windows 7, Windows Server 2008 [R2](#))

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

Type-Bug

Area-UI

Feature-Omnibox

Feature-Protector

Feature-Search

Action-FeedbackNeeded

Mstone-19

 **no search.png**
751 KB [View](#) [Download](#)



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

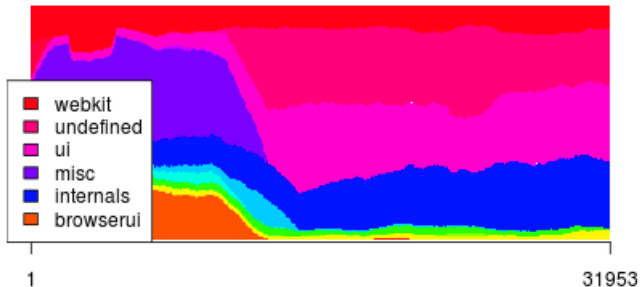
- Alternate items assigned to **dev** and **test**
- Dev sets sizes:

Tracker	Output	# Items	# Labels
Chromium	Subcomponent	31,953	75
Chromium	Assigned	16,154	591
Android	Subcomponent	888	12
Android	Assigned	718	72
Firefox	Assigned	12,733	503
Launchpad	Assigned	18,634	1,970



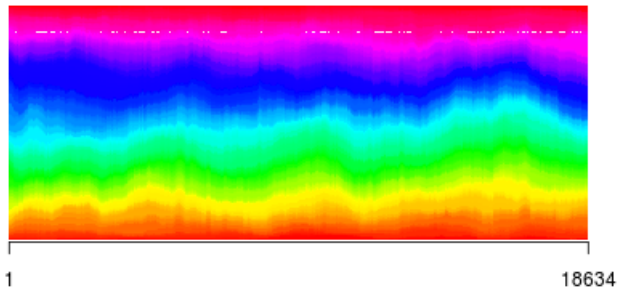
Evolving class distribution

Chromium Subcomponent



Evolving class distribution

Launchpad Assigned



Progressive validation

- For $i = 1$ to ∞
 - ▶ 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

- Tamrawi et al. 2011, Fuzzy set and cache-based approach for bug triaging.
- 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.

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

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

- Adaptive, per-feature learning rate (Duchi et al. 2010, Streeter and McMahan 2010)
- Learning rate larger for infrequent features.



Reduction from multiclass

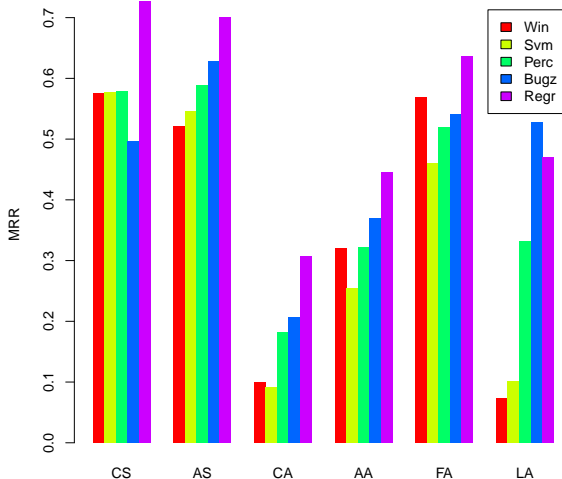
- One-versus-all reduction

$$T(\mathbf{x}, y) = \{(\mathbf{x}', I(y = y')) \mid y' \in Y, x'_{h(i, y')} = x_i\}$$

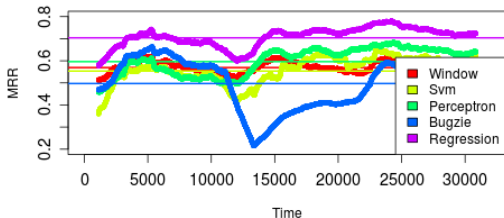
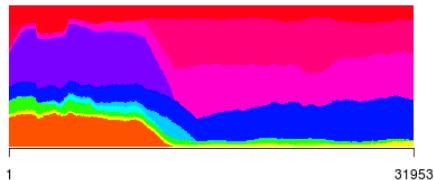
- $h(i, y')$ composes the index i with the label y' by hashing.



Summary of results (test)

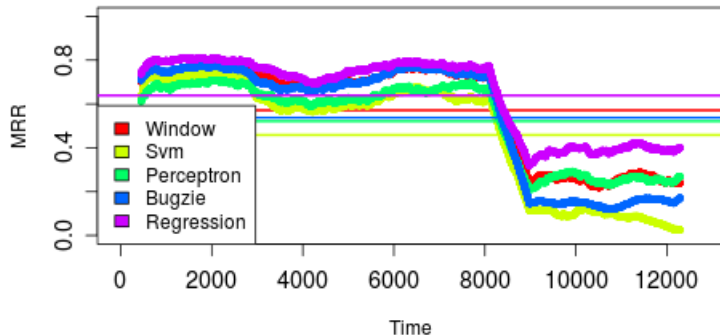


Chromium SUBCOMPONENT



Firefox ASSIGNED

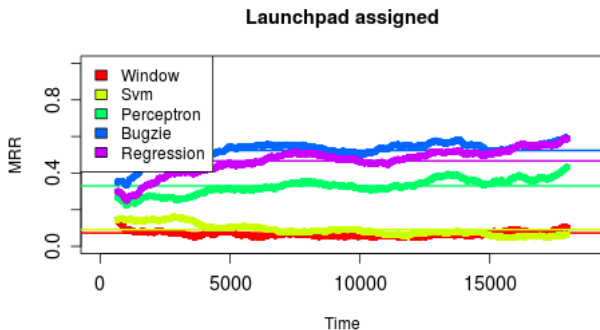
Firefox assigned



- Data becomes more difficult around 9.000



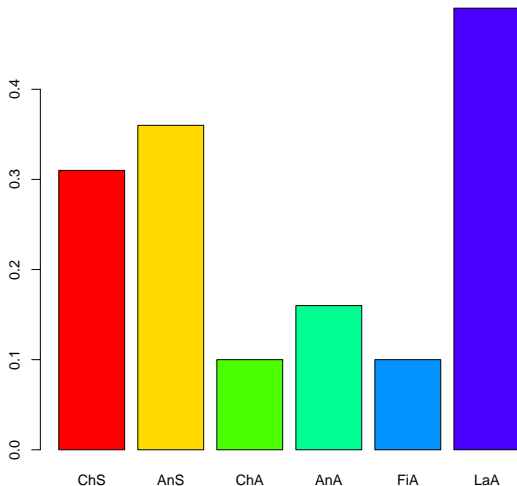
Launchpad ASSIGNED



- Little concept drift
- ≈ 2000 labels: hashing collisions



Best improvement over Window



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**

