# Data Mining: Learning from Large Data Sets - Fall Semester 2015

eholmer@student.ethz.ch
melisr@student.ethz.ch
pmichal@student.ethz.ch

December 23, 2015

## Explore-Exploit tradeoffs in Recommender Systems

### LinUCB

We implemented the LinUCB algorithm like it is given on the last slide of lecture 11 of *Data Mining: Learning from Large Data Sets*.
This means that we recommend the article from the given set of articles $\mathcal{A}_t$ that has the highest **UCB** value:

$$x_t = \underset{x \in \mathcal{A}_t}{\operatorname{argmax}} \mathbf{UCB}_x$$

with

$$\mathbf{UCB}_x = M_x^{-1} b_x z_t + \alpha \sqrt{z_t^T M_x^{-1} z_t}$$

where $M_x$ and $B_x$ are article-specific variables, and $z_t$ is the vector with user features. $\alpha = 0.1875$ gives the best result.
Since inverting a matrix is a time expensive operation, we don't only keep $M_x$ and $b_x$ in memory for each article $x$, but also $M_x^{-1}$ and $M_x^{-1} b_x$. Upon a right recommendation, these variables are updated arrording to

$$M_x \leftarrow M_x + z_t z_t^T \text{ and } b_x \leftarrow b_x + y_t z_t \ .$$

where $y_t$ denotes the reward.

### Other Approaches

To capture global dependence between user and article features, we also tried a hybrid model, but didn't manage to stay within the time boundary of 15 minutes.

## Member contribution

| | |
|---|---|
| Michal Porvazník | Implemented LinUCB |
| Erik Holmer | Tuned parameter |
| Rik Melis | Optimized runtime |