(Machine)Learning with limited labels

Eirini Ntoutsi
(joint work with Vasileios Iosifidis)

Leibniz University Hannover & L3S Research Center

A good conjuncture for ML/DM (data-driven learning)

Data deluge

Machine Learning advances

Computer power

Enthusiasm

More data = Better learning?

Data deluge

Machine Learning advances

- Data is the fuel for ML
- (Sophisticated) ML methods require more data for training

- However, more data does not necessarily imply better learning

More data != Better learning

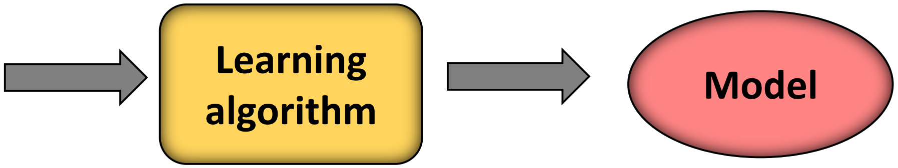
- More data != Better data
- The veracity issue/ data in doubt
  - Data inconsistency, incompleteness, ambiguities, …
- The non-representative samples issue
  - Biased data, not covering the population/problem we want to study
- The label scarcity issue
  - Despite its volume, big data does not come with label information
  - Unlabelled data: Abundant and free
    - E.g., image classification: easy to get unlabeled images
    - E.g., website classification: easy to get unlabeled webpages
  - Labelled data: Expensive and scarce
- …

Why label scarcity is a problem?

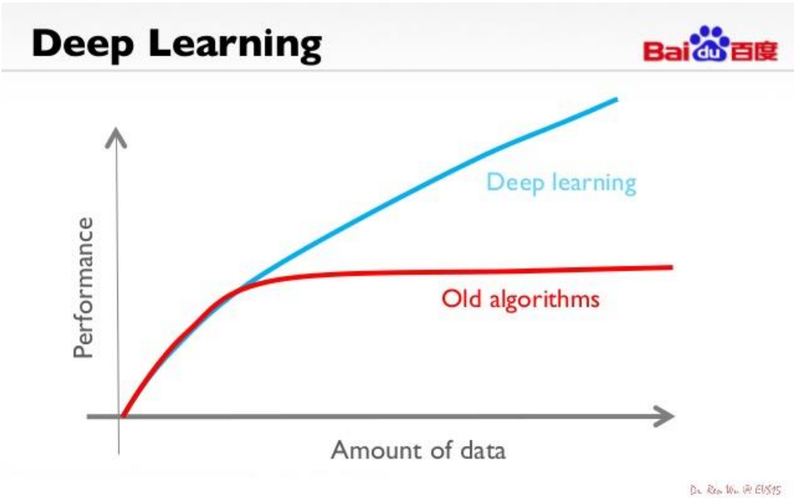
- Standard supervised learning methods will not work

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



- Esp. a big problem for complex models, like deep neural networks.



Source: https://tinyurl.com/ya3svsxb

How to deal with label scarcity?

- A variety of methods is relevant
  - Semi-supervised learning
    - Exploit the unlabelled data together with the labelled one

    This talk!
  - Active-learning
    - Ask the user to contribute labels for a few, useful for learning instances

    Past, ongoing work!
  - Data augmentation
    - Generate artificial data by expanding the original labelled dataset

    Ongoing work!
  - ....

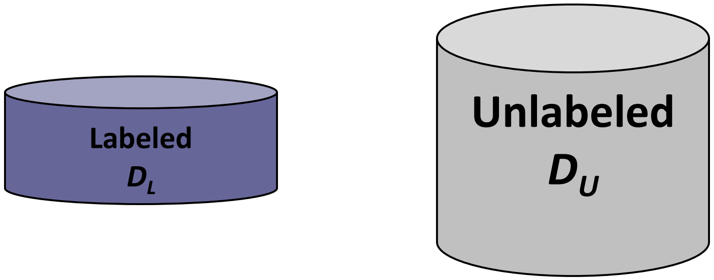
In this presentation

Semi-supervised learning

(or, exploiting the unlabelled data together with the labelled one)

Semi-supervised learning

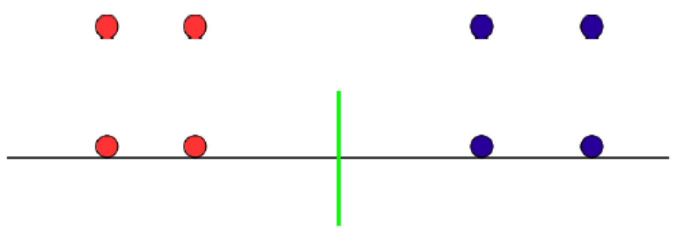
- Problem setting
  - Given: Few initial labelled training data D_L =(X_l,Y_l) and unlabelled data D_U = (X_u)
  - Goal: Build a model using not only D_L but also D_U



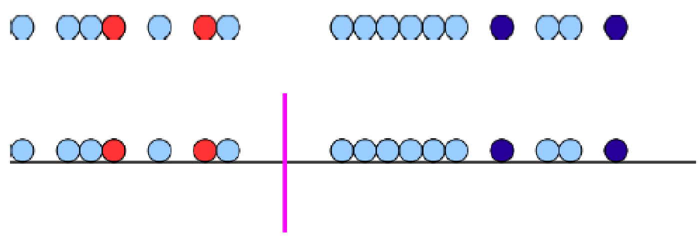
The intuition

- Lets consider only the labelled data
  - We have two classes: red & blue

Important prerequisite: the distribution of examples, which the unlabeled data will help elucidate, should be relevant for the classification problem



- Lets consider also some unlabelled data (light blue)



- The unlabelled data can give a better sense of the class separation boundary (in this case)

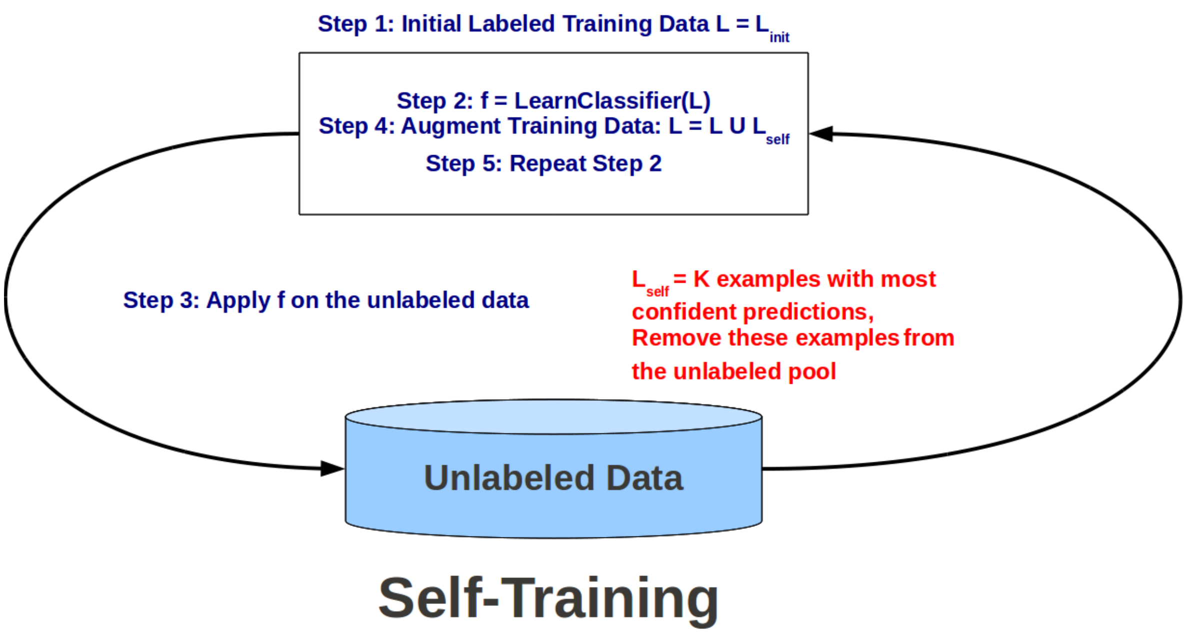
Semi-supervised learning methods

- Self-learning
- Co-training
- Generative probabilistic models like EM

  Not included in this work.
- …

Semi-supervised learning: Self-learning

- Given: Small amount of initial labelled training data D_L
- Idea: Train, predict, re-train using classifier's (best) predictions, repeat

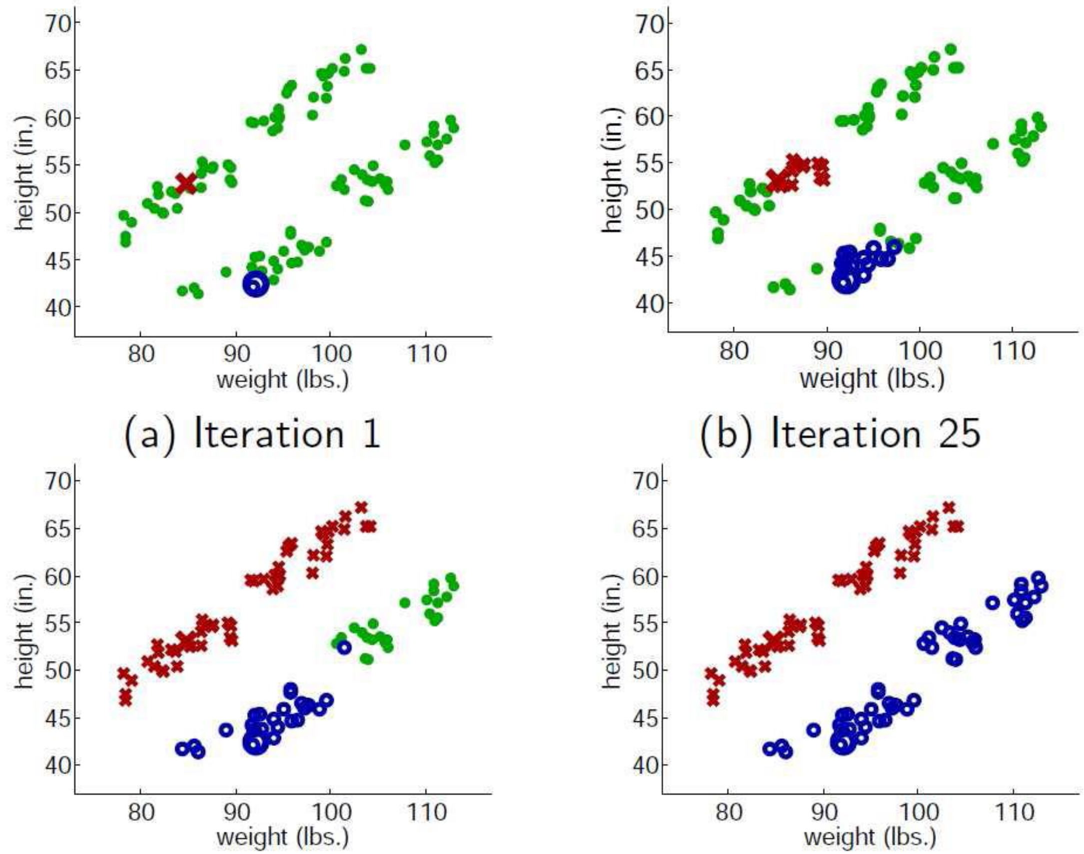


- Can be used with any supervised learner.

Source: https://tinyurl.com/y98clzxb

Self-Learning: A good case

- Base learner: KNN classifier



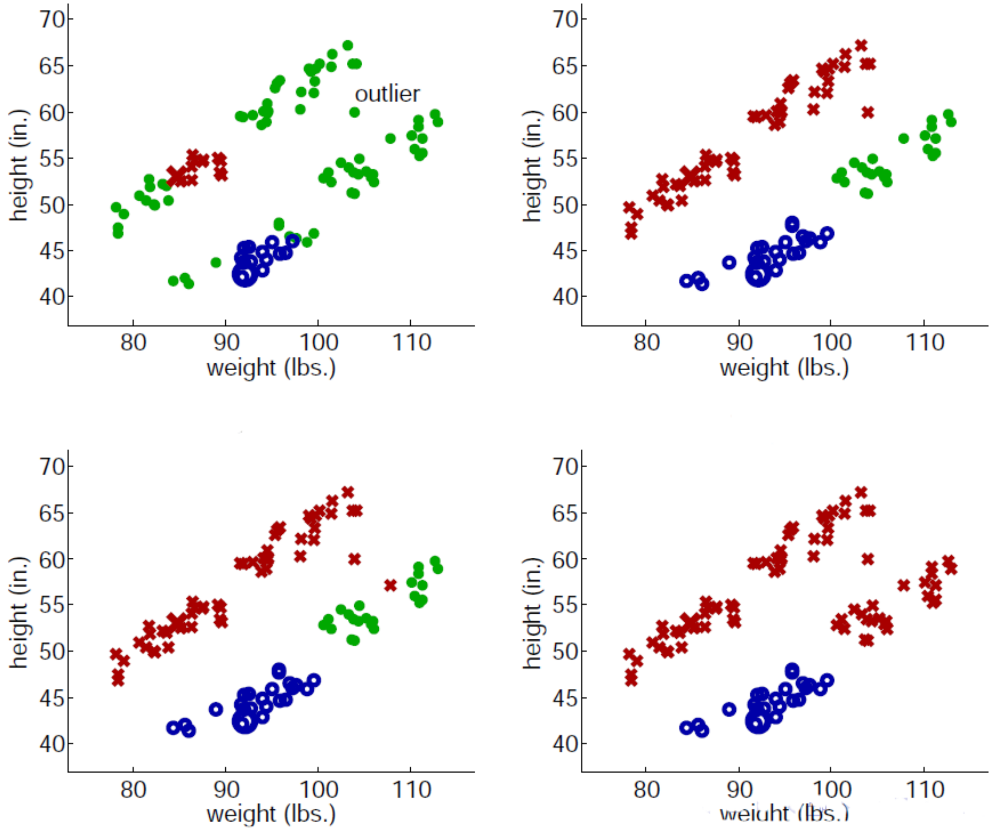
(a) Iteration 1 (b) Iteration 25

(c) Iteration 74 (d) Final labeling of all instances

Source: https://tinyurl.com/y98clzxb

Self-Learning: A bad case

- Base learner: KNN classifier
- Things can go wrong if there are outliers. Mistakes get reinforced.



Source: https://tinyurl.com/y98clzxb

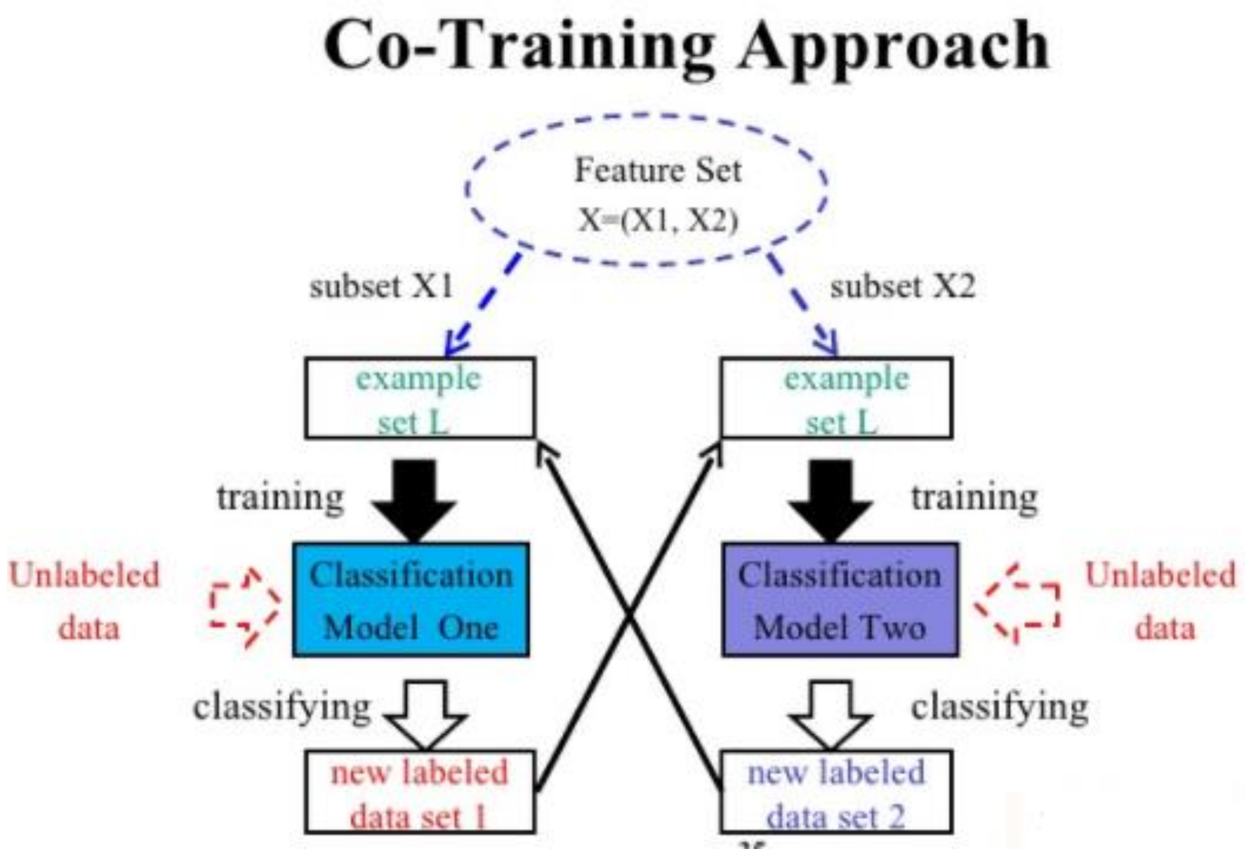
Semi-supervised learning: Co-Training

- Given: Small amount of initial labelled training data
  - Each instance *x*, has two views $x=[x^1, x^2]$
  - E.g., in webpage classification:
    1. Page view: words appearing on the web page
    2. Hyperlink view: words underlined in links pointing in the webpage from other pages

- Co-training utilizes both views to learn better with fewer labels

- Idea: Each view teaching (training) the other view
  - By providing labelled instances

Semi-supervised learning: Co-Training



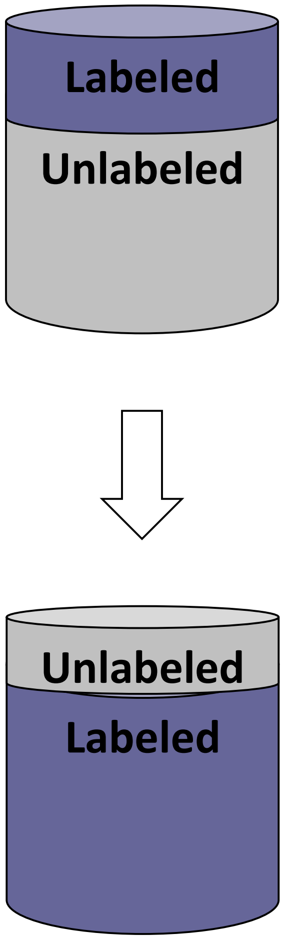
Semi-supervised learning: Co-Training

- Assumption
  - Views should be independent
    - Intuitively, we don’t want redundancy between the views (we want classifiers that make different mistakes)
  - Given sufficient data, each view is good enough to learn from

Self-learning vs co-training

- Despite their differences
  - Co-training splits the features, self-learning does not
- Both follow a similar training set expansion strategy
  - They expand the training set by adding labels to (some of) the unlabeled data.
  - So, the traning set is expanded via: real (unlabeled) instances with predicted labels

  - Both self learning & co-training incrementally uses the unlabeled data.
  - Both self learning & co-training propagate the most confident predictions to the next round



This work

Semi-supervised learning for textual data

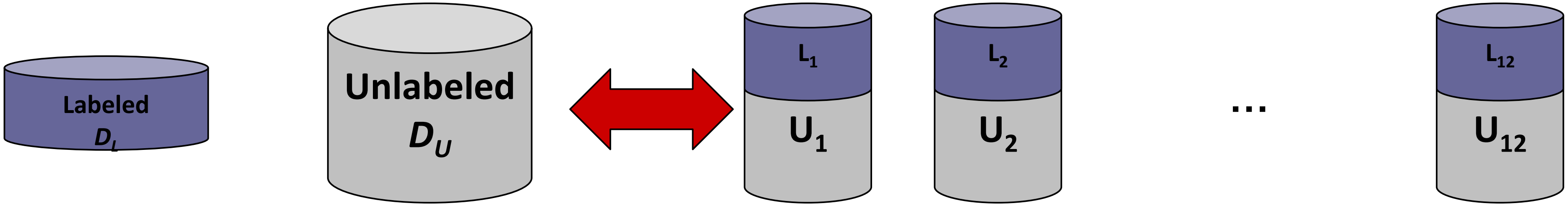
(self-learning, co-training)

The TSentiment15 dataset

- We used self-learning and co-training to annotate a big dataset
  - the whole Twitter corpus of 2015 (228M tweets w.o. retweets, 275M with)
  - The annotated dataset is available at: https://l3s.de/~iosifidis/TSentiment15/
- The largest previous dataset is
  - TSentiment (1,6M tweets collected over a period of 3 months in 2009)
- In both cases, labelling relates to sentiment
  - 2 classes: positive, negative

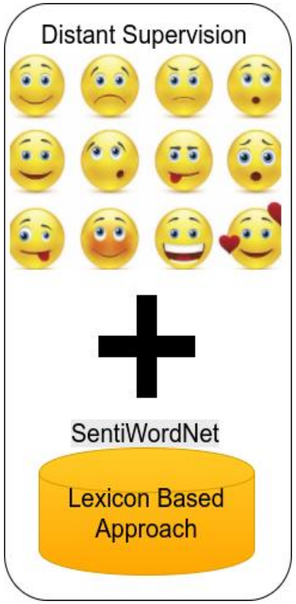
Annotation settings

- For self-learning:
  - the features are the unigrams
- For co-training: we tried two alternatives
  - Unigrams and bigrams
  - Unigrams and language features like part-of-speech tags, #words in capital, #links, #mentions, etc.
- We considered two annotation modes:
  - Batch annotation: the dataset was processed as a whole
  - Stream annotation: the dataset was proposed in a stream fashion



How to build the ground truth (D_L)

- We used two different label sources
  - Distant Supervision
    - Use emoticons as proxies for sentiment
    - Only clearly-labelled tweets (with only positive or only negative emoticons) are kept
  - SentiWordNet: a lexicon-based approach
    - The sentiment score of a tweet is an aggregation of the sentiment scores of its words (the latest comes from the lexicon)

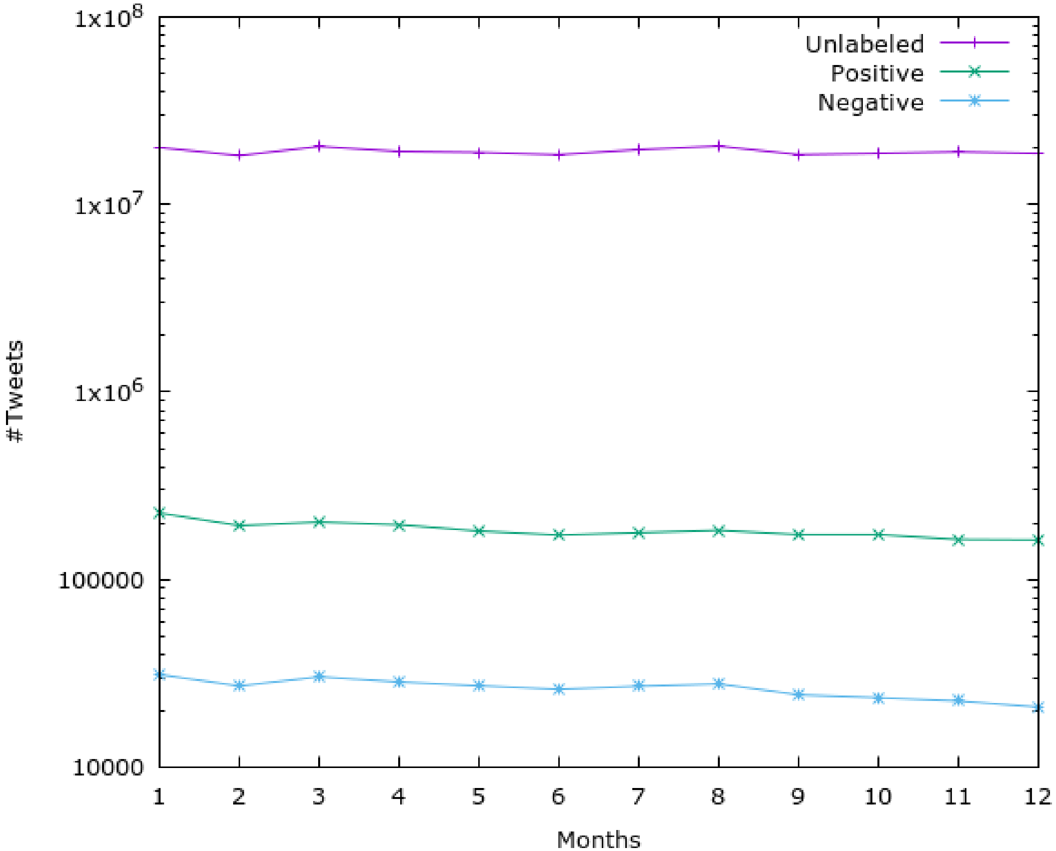


	SWN. Pos.	SWN. Neg.	SentiW. Neutral
Emot. Pos.	2,211,091	840,787	807,887
Emot. Neg.	1,032,536	316,662	157,322

- They agree on ~2,5M tweets → ground truth

Labeled-unlabeled volume (and over time)

- On monthly average, D_U 82 times larger than D_L
- Positive class is overrepresented, average ration positive/negative per month =3



Batch annotation: Self-learning vs co-training

Self –learning

δ	positive predictions	negative predictions	unlabeled
65%	201,860,127 (88.46%)	26,315,605 (11.53%)	**1.13%**
70%	200,212,418 (88.49%)	26,033,446 (11.50%)	**1.97%**
75%	198,296,101 (88.59%)	25,525,791 (11.40%)	**3.02%**
80%	196,017,401 (88.78%)	24,757,934 (11.21%)	**4.34%**
85%	193,134,363 (89.06%)	23,720,362 (10.93%)	**6.03%**
90%	189,271,805 (89.49%)	22,217,878 (10.50%)	**8.36%**
95%	183,012,328 (90.21%)	19,843,802 (9.78%)	**12.10%**
100%	650,450 (99.86%)	877 (0.13%)	**99.71%**
Initial Model	2.211.091 (87,47%)	316.662(12,52%)	

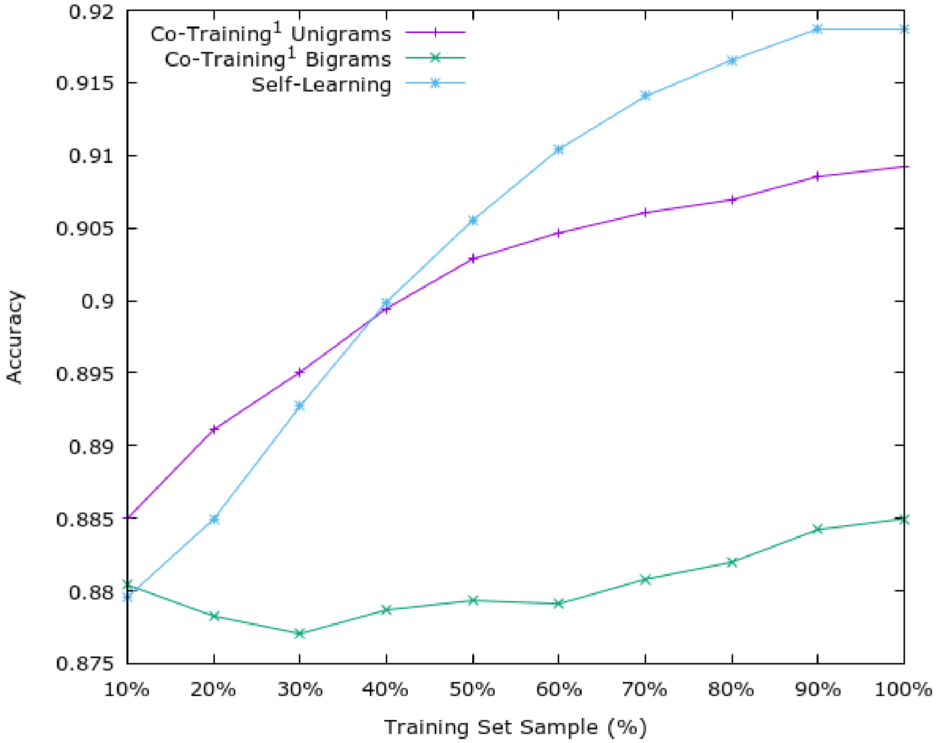
Co-training

δ	positive predictions	negative predictions	unlabeled
65%	175,704,567 (76.64%)	53,547,361 (23.35%)	0.66%
70%	178,361,861 (78.26%)	49,544,295 (21.73%)	1.25%
75%	180,646,395 (79.90%)	45,419,649 (20.09%)	2.04%
80%	182,180,488 (81.52%)	41,287,186 (18.47%)	3.17%
85%	182,758,504 (83.04%)	37,300,375 (16.95%)	4.65%
90%	182,707,849 (85.06%)	32,069,200 (14.93%)	6.93%
95%	179,527,239 (87.43%)	25,810,993 (12.56%)	11.02%
100%	1,281,748 (99.60%)	5,116 (0.39%)	99.44%
Initial Model	2.211.091 (87,47%)	316.662(12,52%)	

- The more selective δ is the more unlabeled tweets
- The majority of the predictions refer to positive class
- The model is more confident on the positive class

- Co-training labels more instances than self-learning
- Co-training learns the negative class better than self-learning

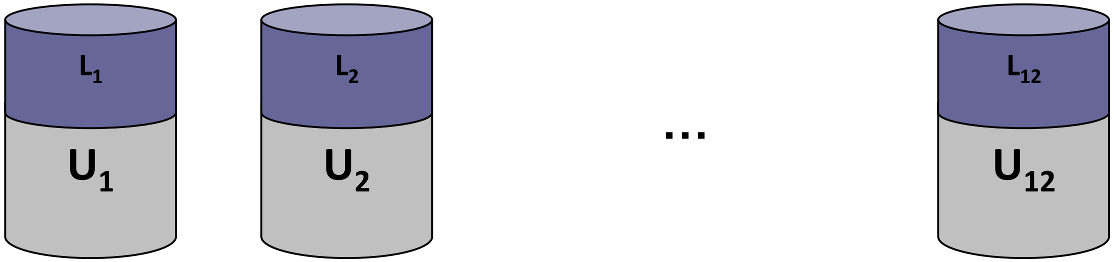
Batch annotation: Effect of labelled set sample



- When the number of labels is small, co-training performs better
- With >=40% of labels, self-learning is better

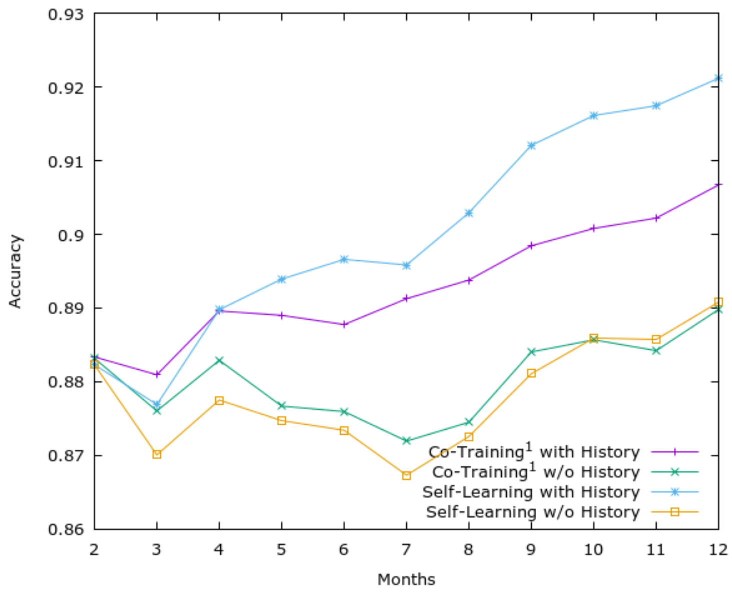
Stream annotation

- Input: stream in monthly batches: $((L_1, U_1), (L_2, U_2), \ldots, (L_{12}, U_{12}))$
- Two variants are evaluated, for training:
  - Without history: We learn a model on each month i (using L_i, U_i).
  - With history: For a month i, we consider as $L_i = \sum_{i=1}^{i} L_i$. Similarly for U_i.
- Two variants also for testing:
  - Prequential evaluation: use the L_{i+1} as the test set for month i
  - Holdout evaluation: we split D into D_{train}, D_{test} . Training/ testing similar to before but only on data from D_{train}, D_{test}, respectively.



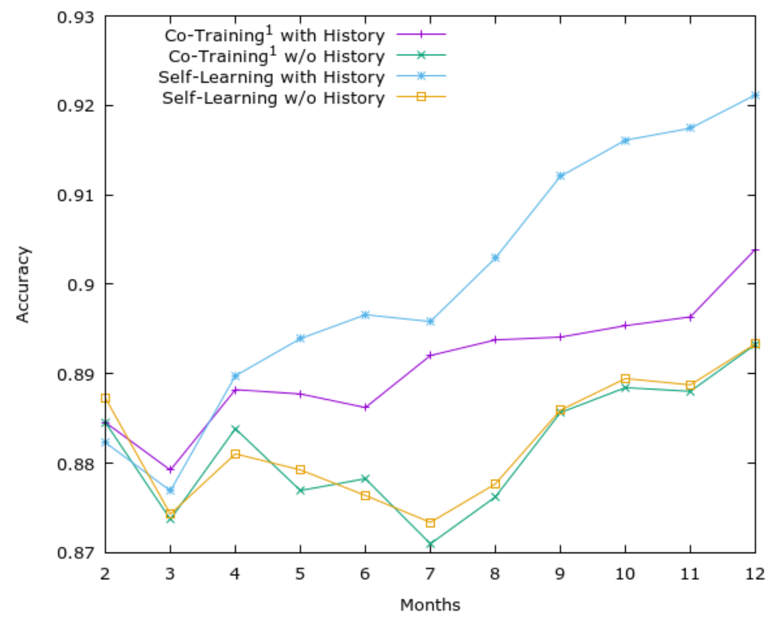
Stream: Self-learning vs co-training

Prequential



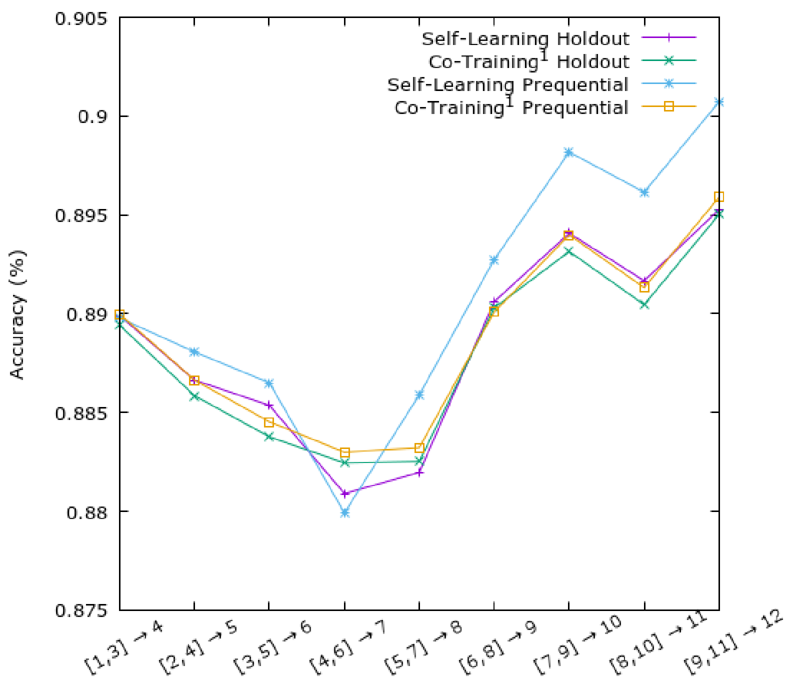
- History improves the performance
- For the models with history, co-training is better in the beginning but as the history grows self-learning wins

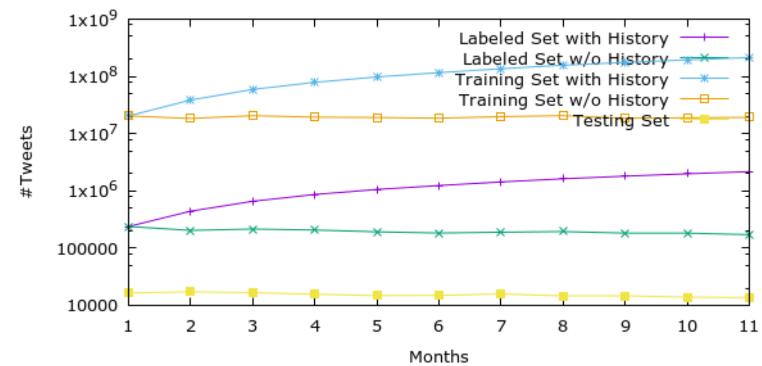
Holdout

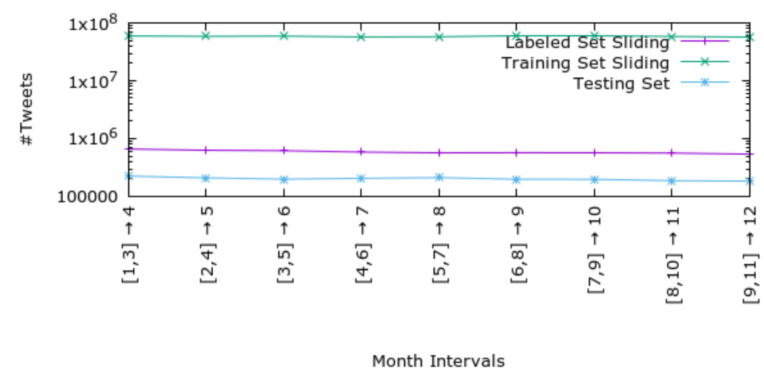


Stream: the effect of the history length

- We used a sliding window approach
  - E.g., training on months [1-3] using both labeled and unlabeled data, test on month 4.
  - Small decrease in performance comparing to the full history case but much more light models







Class distribution of the predictions

- Self-learning produces more positive predictions than co-training
- Version with retweets results in more balanced predictions
  - Original class distribution w.o. retweets: 87%-13%
  - Original class distribution w. retweets: 75%-25%

δ	Self-Learning noRts	Co-Training noRts	Self-Learning with Rts	Co-Training with Rts
65%	1:8	1:4	1:2	1:2
70%	1:8	1:4	1:2	1:3
75%	1:8	1:4	1:2	1:2
80%	1:8	1:4	1:2	1:2
85%	1:8	1:5	1:2	1:2
90%	1:9	1:6	1:2	1:2
95%	1:9	1:7	1:2	1:2
100%	1:741	1:248	1:2	1:14

Summary

- We annotated a big dataset with semi-supervised learning
  - Self-training
  - Co-training
  - When the number of labels is small, co-training performs better

- Batch vs stream annotation
  - History helps (but we don't need to keep the whole history, a sliding window based approach is also ok)
- Learning with redundancy (retweets)
  - Better class balance in the predictions when retweets are used (because the original dataset is balanced)

Ongoing work

- Thus far: Semi-supervised learning which focuses on label scarcity
- Another way to get around lack of data is data augmentation
  - i.e., increasing the size of the training set by generating artificial data based on the original labeled set
- Useful for many purposes
  - Deal with class imbalance, create more robust models etc
- We investigate different augmentation approaches
  - At the input layer
  - At the intermediate layer
- And how to control the augmentation process
  - The goal is to generate *plausible data* that help with *the classification task*

Thank you for you attention!

Questions/ Thoughts?

- Relevant work
  - V. Iosifidis, E. Ntoutsi, "*Large scale sentiment annotation with limited labels*", KDD, Halifax, Canada, 2017
- TSentiment15 available at:
  - https://l3s.de/~iosifidis/TSentiment15/

www.kbs.uni-hannover.de/~ntoutsi/
ntoutsi@l3s.de