Semi-Supervised Learning and Domain Adaptation in Natural Language Processing

3 Semi-Supervised Learning

3.1 Wrapper Methods

3.1.1 Self-training

3.1.2 Co-training

3.1.3 Tri-training

3.1.4 Soft Self-training, EM and co-EM

小野寺喜行

Semi-Supervised Learning

- Exploiting the marginal distribution of unlabeled data.
- Better models than with labeled data alone.
- This also often leads to degradation in performance.

Sometimes

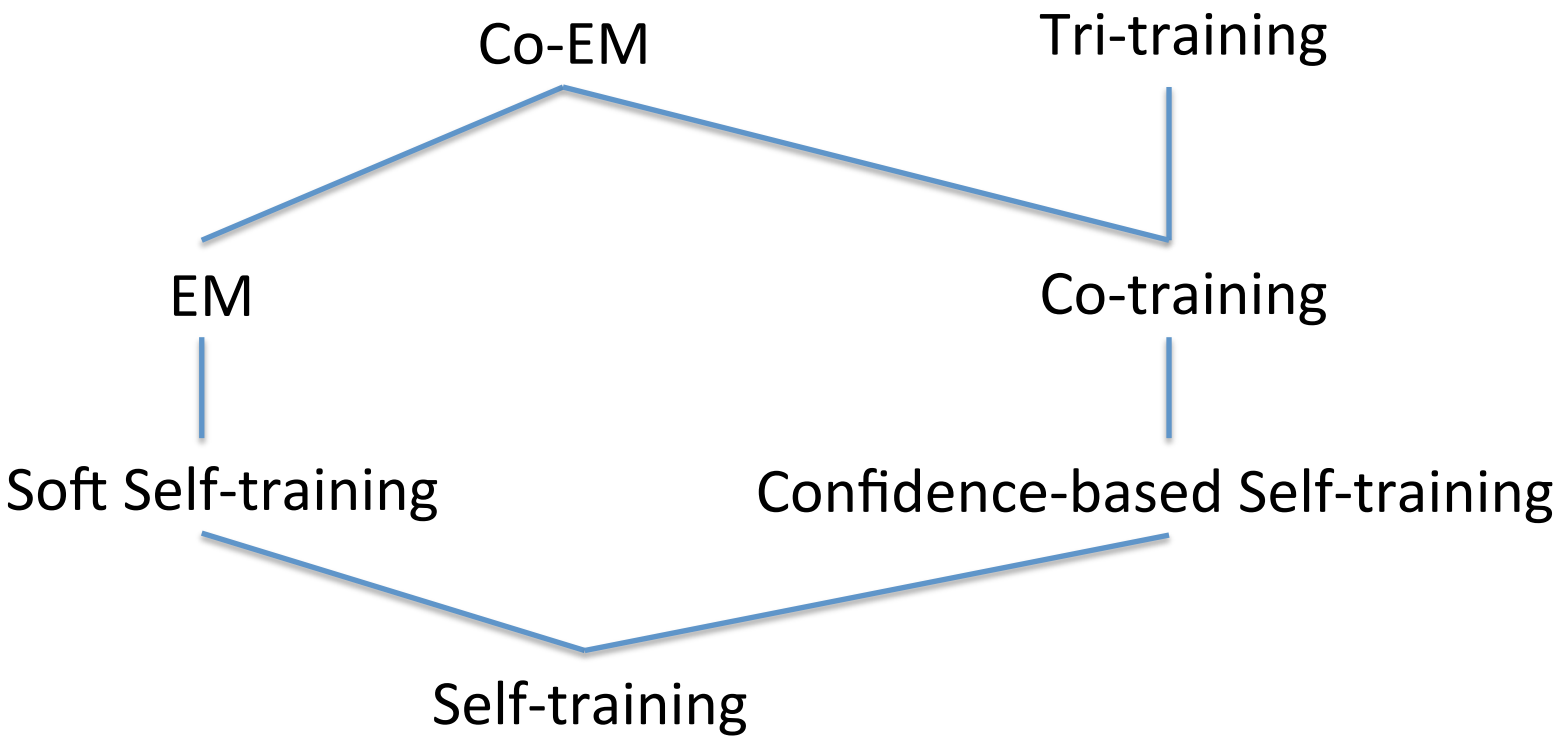
due to invalid assumptions

[Example]

Assumption: Decision boundaries run through sparse regions

Poorly performance on data generated
by two heavily overlapping Gaussians.

Wrapper Methods



Self-training

1: $L = \left\{ \left\langle y_i, x_i \right\rangle \right\}_{i=1}^{N} \quad U = \left\{ x_i \right\}_{i=1}^{M}$

2: $c \leftarrow$ train(L)

3: **while** stopping criterion is not met **do**

4: $L \leftarrow L$ + select(label(U,c))

5: $c \leftarrow$ train(L)

6: **end while**

7: **return** c

Self-training

1: $L = \left\{ \left\langle y_i, x_i \right\rangle \right\}_{i=1}^{N} \quad U = \left\{ x_i \right\}_{i=1}^{M}$

2: $c \leftarrow$ train(L)

c : classifier L : a set of labeled data points

U : a large volume of unlabeled data

Self-training

3: **while** stopping criterion is not met **do**

Stopping criterion :

Held-out data, or cross-validation, is typically used to estimate a reasonable number K of fixed rounds.

To make self-training more robust

Throttling : we only select K data points in each pass over the unlabeled data

Balancing : using a select function that selects only the K most confidently labeled data points in each class

Pooling : always using a (randomly sampled) subset of the unlabeled data

Self-training

4: $L \leftarrow L$ + select(label(U,c))

Select function :

select only a subset of the newly labeled data

Common strategy

Import only data points that are labeled,
say with confidence of more than 90%.

Delible self-training:

delete previously imported data in each round of self-training

Co-training

1: $L = \left\{ \left\langle y_i, x_i \right\rangle \right\}_{i=1}^{N} \quad U = \left\{ x_i \right\}_{i=1}^{M}$

2: $c \leftarrow$ train(L)

3: **while** stopping criterion is not met **do**

4: $\quad c_1 \leftarrow$ train(view$_1$(L))

5: $\quad c_2 \leftarrow$ train(view$_2$(L))

6: $\quad L \leftarrow L$ + select(label(U,c_1)) + select(label(U,c_2))

7: **end while**

8: $c \leftarrow$ train(L)

9: **return** c

Tri-training

1: $L = \{\langle y_i, x_i \rangle\}_{i=1}^{N}$ $U = \{x_i\}_{i=1}^{M}$
2: **for** $i \in \{1..3\}$ **do**
3: $c_i \leftarrow \text{train}_i(L)$
4: **end for**
5: **while** stopping criterion is not met **do**
6: **for** $i \in \{1..3\}$ **do**
7: $L_i \leftarrow \emptyset$
8: **for** $\mathbf{x} \in U$ **do**
9: **if** $c_j(\mathbf{x}) = c_k(\mathbf{x})(j,k \neq i)$ **then**
10: $L_i \leftarrow L_i \cup \{\langle c_j(\mathbf{x}), \mathbf{x} \rangle\}$
11: **end if**
12: **end for**
13: $c_i \leftarrow \text{train}_i(L \cup L_i)$
14: **end for**
15: **end while**
16: **return** c_1

Soft Self-training

- Assign weights to the newly labeled data reflecting our confidence in the labeling.
- Add two copies of each newly labeled data point $\mathbf{x}$.

(1)positive class (2)negative class

$\hat{P}(1 \mid \mathbf{x})$ $\hat{P}(0 \mid \mathbf{x})$

Estimated probability

EM

- Run over all unlabeled data in each round

  (instead of pooling batches of unlabeled data)
- Delible soft self-training