Applied Machine Learning
Lecture 14-2: Semi-supervised learning, multi-view learning

Selpi (selpi@chalmers.se)

The slides are further development of Richard Johansson's slides

March 10, 2020

Overview

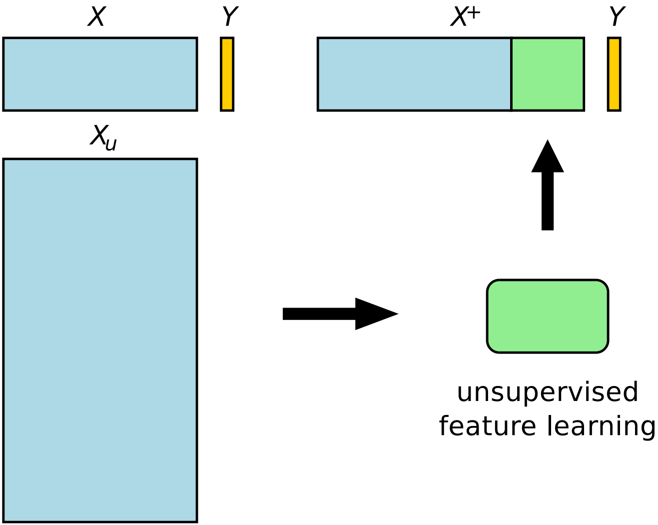
Semi-supervised Learning

Review of this lecture

Semi-supervised learning

- Learns from both labelled and unlabelled data
- Main approaches:
  - To add new features
  - To train the system to label new instances automatically

adding new features

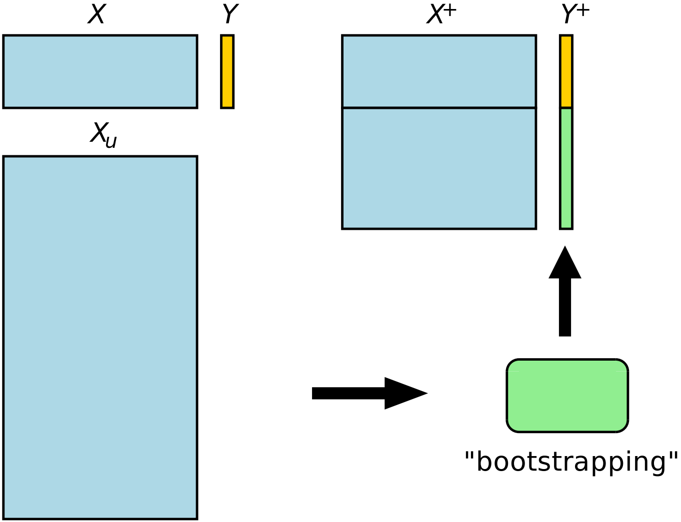


example (text processing)

- Derive **word clusters**; most NLP now uses **word embeddings** that are pre-trained on a large volume of text.

cluster 1000010010111:		cluster 10111111100011:	
time-consuming	225	maine	1758
repugnant	234	turkey	1796
unnerving	240	manhattan	1860
objectionable	243	boston	3704
anticlimactic	244	florida	3764
reprehensible	258	chicago	4535
anti-climactic	270	london	8383
deceiving	289	paris	6329
disrespectful	299	heaven	5864
dissapointing	308	california	7094

generalizing from instances



Bootstrapping methods

A widely used SSL bootstrapping algorithm:
self-training

Self-training

- Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.

  labeled data

  unlabeled data

- Procedure:

1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat:
3. Train f from L using supervised learning.
4. Apply f to the unlabeled instances in U.
5. Remove a subset S from U; add $\{(\mathbf{x}, f(\mathbf{x}))|\mathbf{x} \in S\}$ to L.

Self-training

- Procedure:

1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat:
3. Train f from L using supervised learning.
4. Apply f to the unlabeled instances in U.
5. Remove a subset S from U; add $\{(\mathbf{x}, f(\mathbf{x}))|\mathbf{x} \in S\}$ to L.

- Parameters, e.g., iterations, pool/growth size, select

- Questions:

Q1: This is called a wrapper method. Why?
Q2: Why might this help to build a better system?
Q3: What might go wrong?

Self-training: Summary

Q1: Wrapper?	choice of f left open
Q2: Works when?	broad margin, expected low error
Q3: Limitations?	errors get reinforced
Variants?	Yes, many, e.g., delible self-training, weigh instances,…

Another way of doing self-training

- Use a weighted multi-objective function
  - Train on both labelled and unlabelled data
  - Use lower weight for the unlabelled data that are labelled by the trained model

Co-training

- Different than self-training
- Two different views of the data are used to build the model
- Two different feature sets that provide different complementary information about the instance are needed

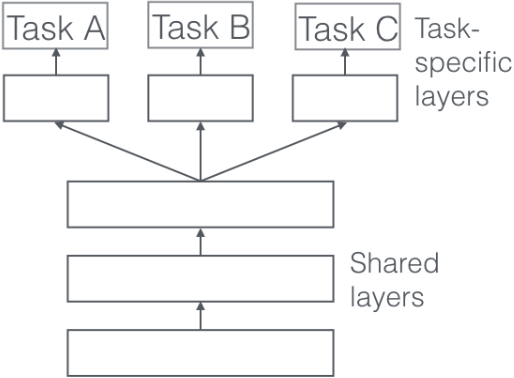
more exotic types of supervision

- in the real world, getting useful training data is often a major bottleneck
- any way to use the data we can get our hands on is welcome

distant supervision

- use an automatic procedure to generate "proxy" training data that **resembles** our real task
- for instance, for sentiment analysis in Twitter, people have used the presence of emoticons as a proxy task
  - presence of `:)` `:-)` `:D` ... ⇒ **positive**
  - presence of `:(` `:/` `:-(` ... ⇒ **negative**

multitask learning



[source]

Overview

Semi-supervised Learning

Review of this lecture

Review of this lecture

- Explain the motivation of semi-supervised learning!
- Could we augment our labelled data using semi-supervised approach? What are the pros and cons of doing this?

Next lecture

- Recap of some concepts from all lectures to reinforce learning