Semi-supervised Learning

Dong Gong
University of Adelaide

Slides by Lingqiao Liu and Dong Gong

Outlines

- Overview of Semi-supervised Learning
- Some commonly used semi-supervised learning approaches
  - Self-training or Pseudo labeling
  - Co-training
  - S3SVM
  - Graph-based approach
- Deep semi-supervised learning
  - Why deep semi-supervised learning
  - Example: Consistency-based approaches

Semi-supervised learning

- What is semi-supervised learning?
  - Learn a model on two types of training data, one with label and one without

- Why bother?
  - Labeled data can be hard to get
    - Human annotation is boring
    - Labeling may require experts or special devices
  - Unlabeled data is cheap

Example

Task: speech analysis

- Switchboard dataset
- telephone conversation transcription
- 400 hours annotation time for each hour of speech

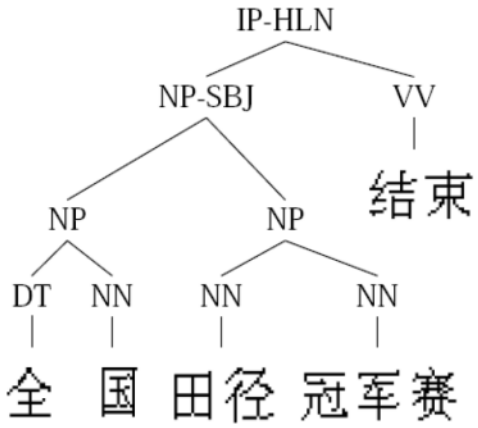
film $\Rightarrow$ `f ih_n uh_gl_n m`

be all $\Rightarrow$ `bcl b iy iy_tr ao_tr ao l_dl`

Example

Task: natural language parsing

- Penn Chinese Treebank
- 2 years for 4000 sentences



"The National Track and Field Championship has finished."

Example

Image classification
Image categorization of “eclipse”

It may not be difficult to label many instances for some tasks.
But there are always more unlabeled data.

Semi-supervised learning

- Goal:
  - Using both labeled and unlabeled data to build better models, than using each one alone
- Notations:

input instance x, label y

learner $f : \mathcal{X} \mapsto \mathcal{Y}$

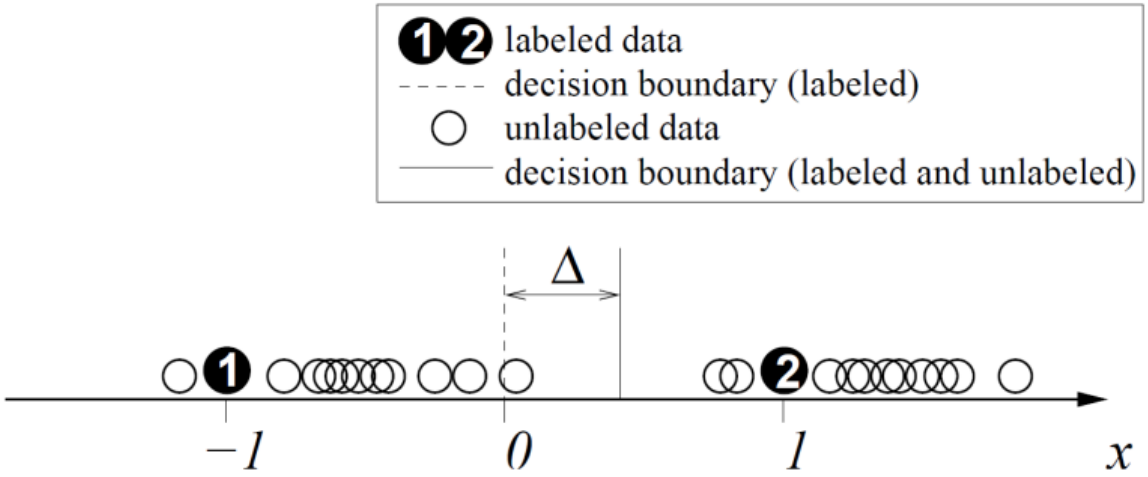
labeled data $(X_l, Y_l) = \{(x_{1:l}, y_{1:l})\}$

unlabeled data $X_u = \{x_{l+1:n}\}$, available during training

usually $l \ll n$

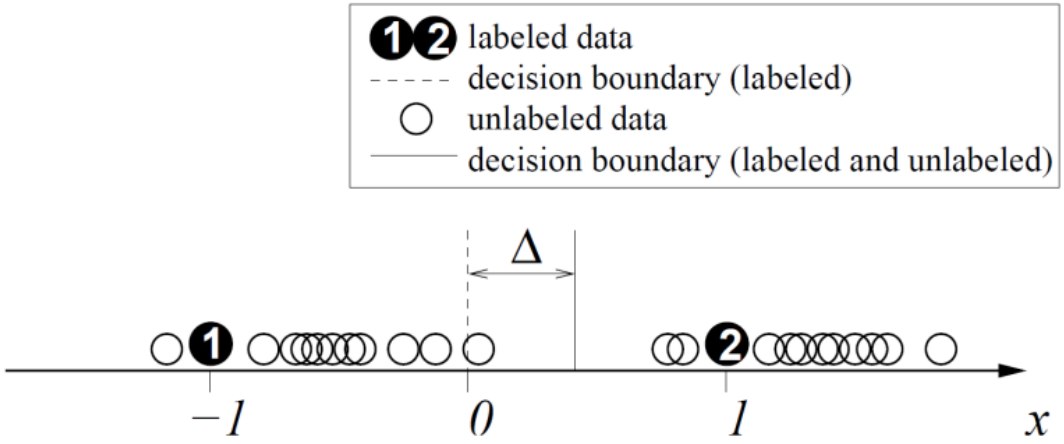
test data $X_{test} = \{x_{n+1:}\}$, not available during training

How can unlabeled data ever help?



- We usually have some assumptions about the data distribution of each class in the unlabeled dataset
- Example assumption: each class is a coherent group (e.g. Gaussian)
  - In the above figure, with and without unlabeled data: decision boundary shift

How can unlabeled data ever help?



- With and without unlabeled data: decision boundary shift.
  - The unlabeled data points shift the decision boundary (as a more accurate boundary).
- Example assumption: each class is a coherent group (e.g. Gaussian)

This is just one of many ways to use unlabelled data; Different Semi-supervised learning approach may take different assumptions

Does unlabeled data always help?

- Unfortunately, this is not the case.
  - The simple assumption may not hold

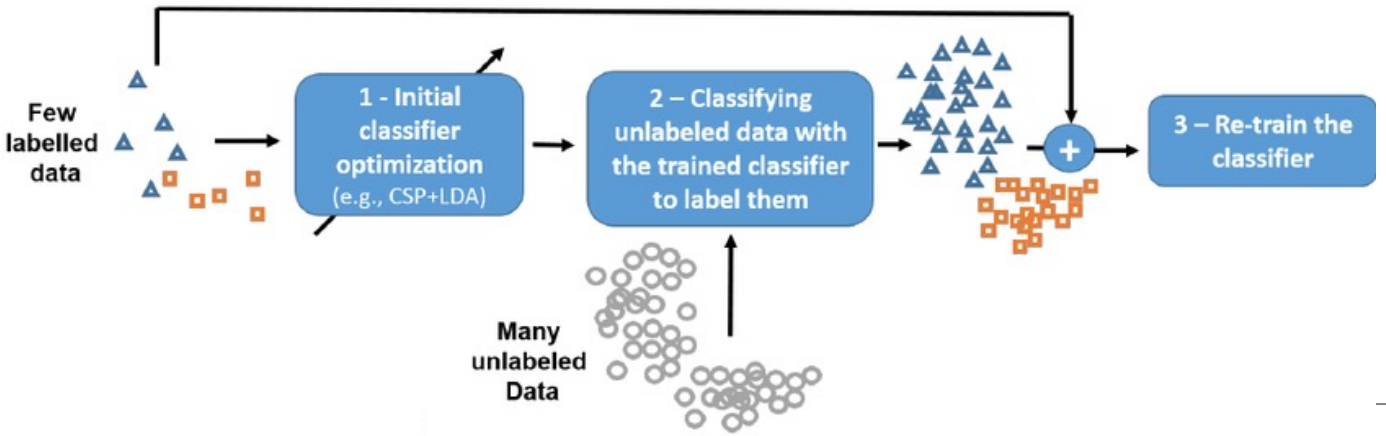
Semi-supervised learning approach

- Many of them:
  - Self-training or Pseudo-label-based approaches
  - Co-training
  - Tri-training
  - Semi-supervised Support Vector Machine (S3VM)
  - …

- An active research direction
- This lecture: some classic methods and simple methods, but can be very useful in practise.

Self-training

- Also called Pseudo labeling approach
  - Assigning unbaled samples pseudo label in learning process
- Algorithm
  - Train f from (X_l, Y_l)
  - Predict on $x \in X_u$, get $\hat{y}$
  - If the prediction is sufficiently confident, add $(x, \hat{y})$ to labeled data
  - Repeat



Prediction confidence

- How to calculate the confidence of prediction
  - Many possible ways
- The classification probability from softmax can be used to measure the confidence on the predicted labels.
  - For example:
    - Three classes
    - A sample is classified as the #2nd class with
      - 0.85 ([0.03, 0.85, 0.12]) → high confidence
      - 0.4 ([0.3, 0.4, 0.3]) → low confidence
    - A threshold may be needed to define "high confidence".

$$P(\hat{\mathbf{y}} = i|\mathbf{x}) = softmax(\mathbf{W}\mathbf{x} + \mathbf{b}) = \frac{e^{\mathbf{w}_i\mathbf{x}+\mathbf{b}}}{\sum_j e^{\mathbf{w}_j\mathbf{x}+\mathbf{b}}}$$

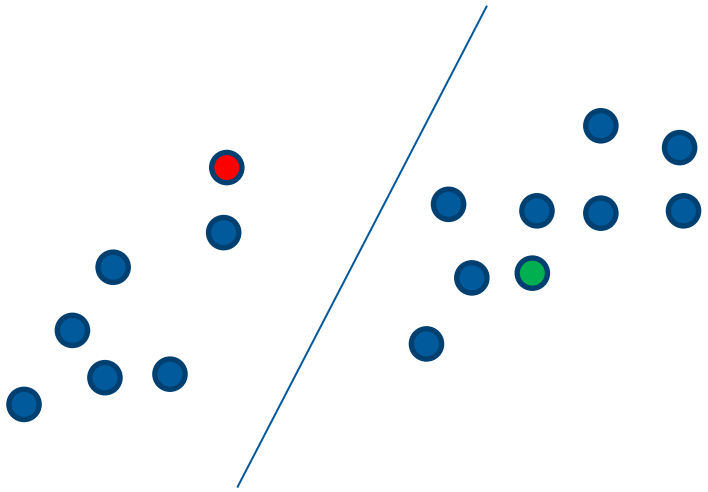
e.g. $[-1, 2, -0.4] \rightarrow [0.0437, 0.8768, 0.0795]$

Self-training

- Assumption:
  - High confidence predictions are correct
- Why it helps?
  - Correct prediction does not mean zero loss value.
  - Optimizing on the confident pseudo samples to further lifting the performance on these samples.
  - More samples -> better model -> more accurate prediction -> More samples

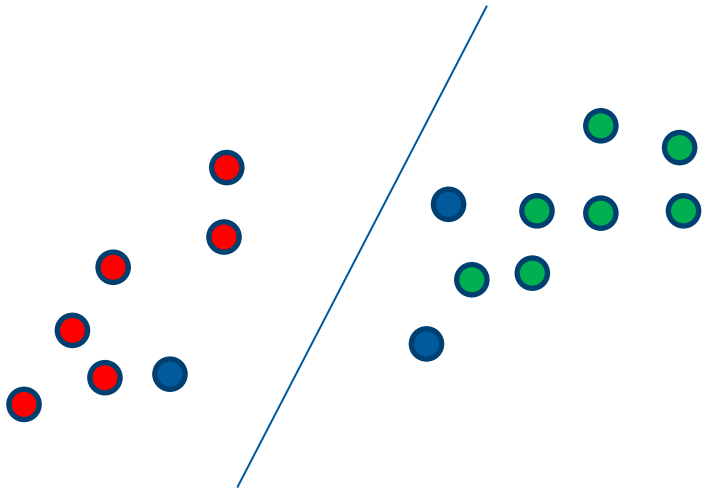
Self-training: Example

Training results on the labeled samples.



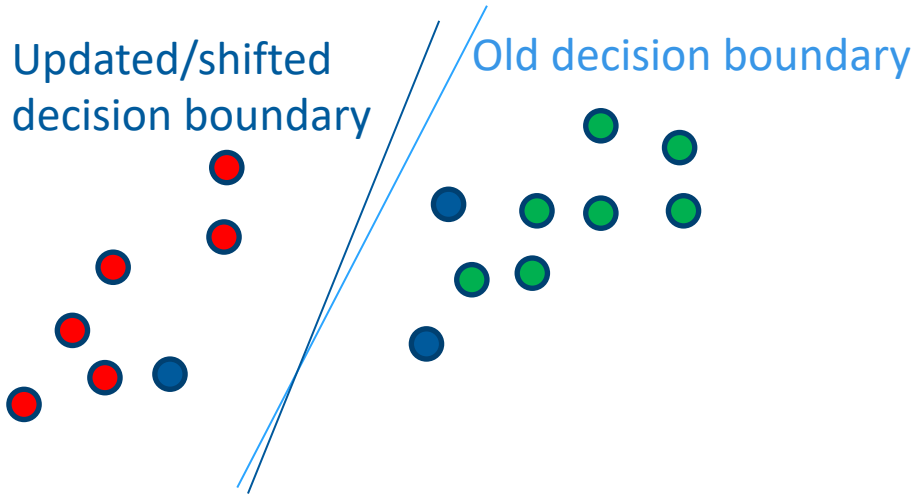
Self-training: Example

Assigning pseudo labels.



Self-training: Example

Re-train the model on the labeled data and data with pseudo labels.

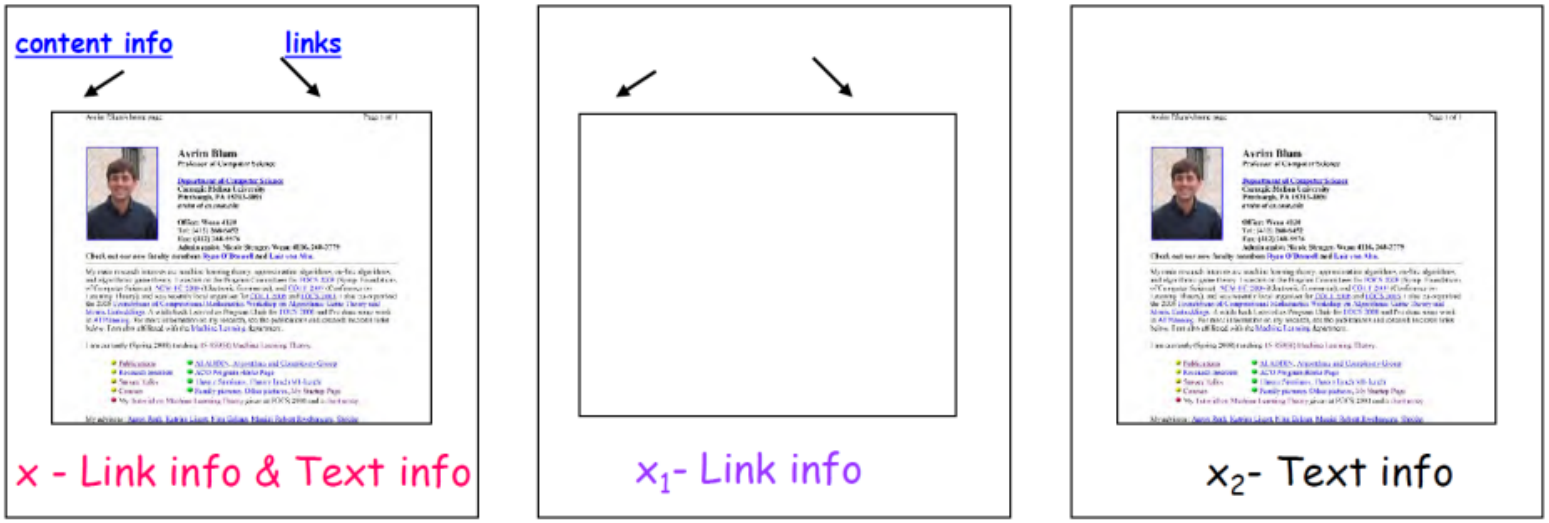


Self-training

- Advantages:
  - Simple wrapper approach: apply to any model flexibly.
  - Easy to implement
- Disadvantages
  - If Pseudo labels are incorrect, no way to correct it.
  - The early mistakes can reinforce themselves.
  - Sensitive to prediction error

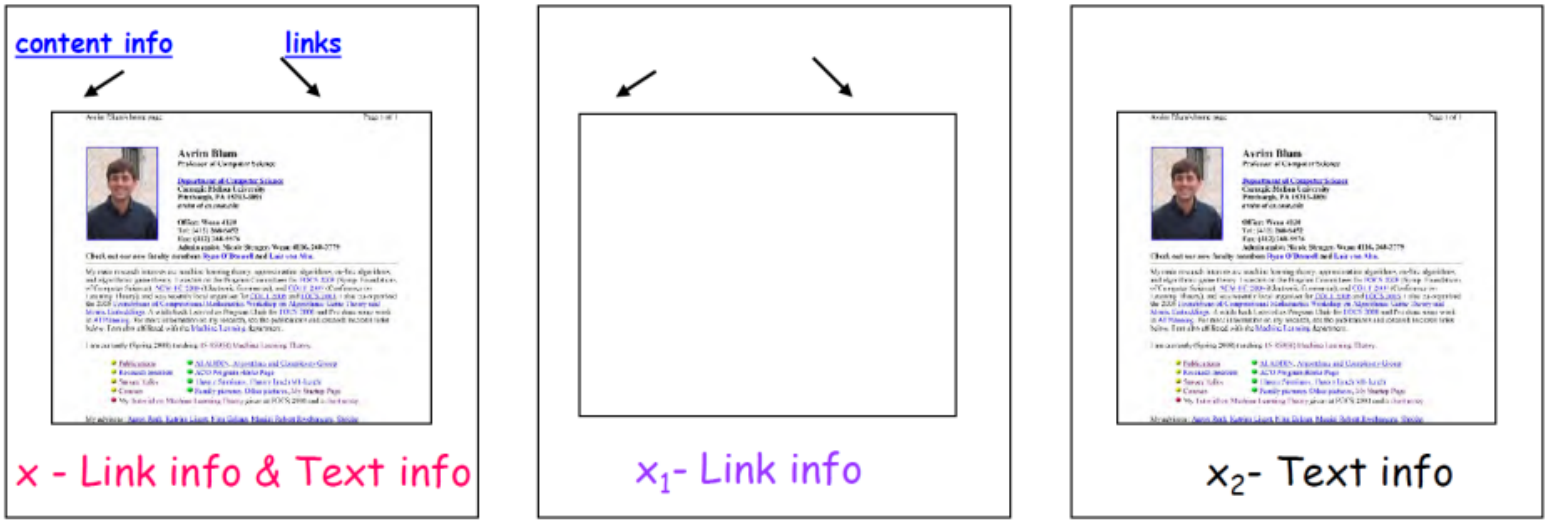
Co-training

- Idea: many problems have two different sources of info you can use to determine label
- E.g., classifying webpages: can use words/images/contents on page or words on links pointing to the page



Co-training

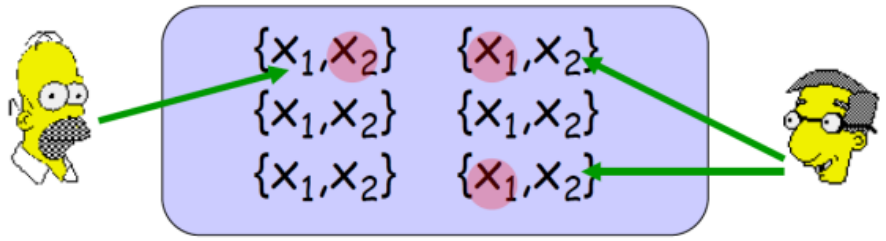
- e.g., “colleagues” pointing to a page is a good indicator it is a faculty home page.
- e.g., “I am teaching ML course” on a page is a good indicator it is a faculty home page.



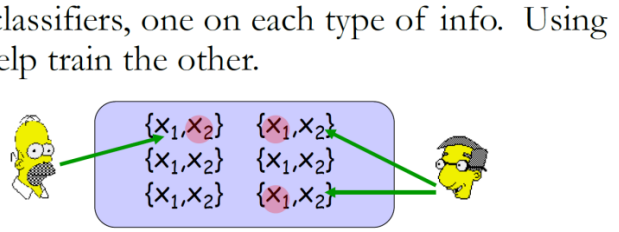
Co-training

- Then look for unlabelled examples where one rule is confident and the other is not. Have it label the example for the other.
- For example, if the prediction from one classifier is sufficient confident, it generates pseudo label for the other classifier

Training 2 classifiers, one on each type of info. Using each to help train the other.



Co-training



- Feature split

  Each instance is represented by two sets of features $x = [x^{(1)}; x^{(2)}]$
  - $x^{(1)}$ = image features
  - $x^{(2)}$ = web page text
  - This is a natural feature split (or multiple views)

  Co-training idea:
  - Train an image classifier and a text classifier
  - The two classifiers teach each other

- Co-training algorithm

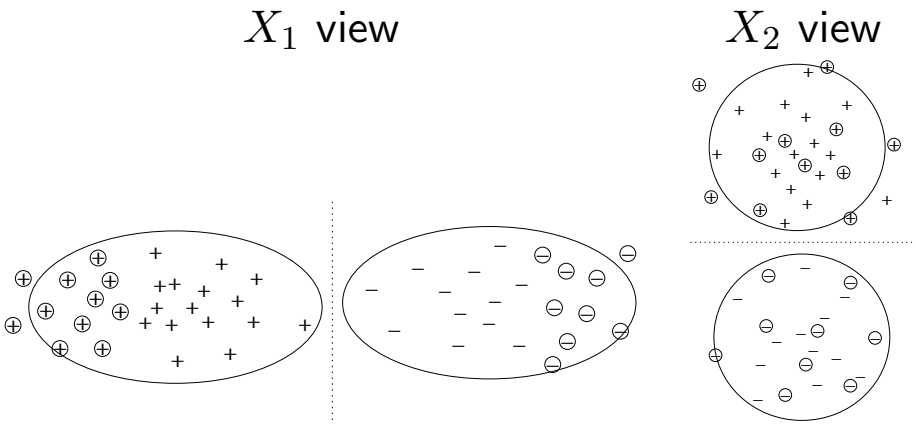
  1. Train two classifiers: $f^{(1)}$ from $(X_l^{(1)}, Y_l)$, $f^{(2)}$ from $(X_l^{(2)}, Y_l)$.
  2. Classify X_u with $f^{(1)}$ and $f^{(2)}$ separately.
  3. Add $f^{(1)}$'s k-most-confident $(x, f^{(1)}(x))$ to $f^{(2)}$'s labeled data.
  4. Add $f^{(2)}$'s k-most-confident $(x, f^{(2)}(x))$ to $f^{(1)}$'s labeled data.
  5. Repeat.

Co-training

- Assumptions for feature splitting-based co-training

Assumptions

- feature split $x = [x^{(1)}; x^{(2)}]$ exists
- $x^{(1)}$ or $x^{(2)}$ alone is sufficient to train a good classifier
- $x^{(1)}$ and $x^{(2)}$ are conditionally independent given the class



Co-training

- How to apply co-training if there is only one source?
  - Generate two independent classifiers: e.g. SVM with different kernels, two different neural networks
  - Key: make sure those classifiers make independent decisions.

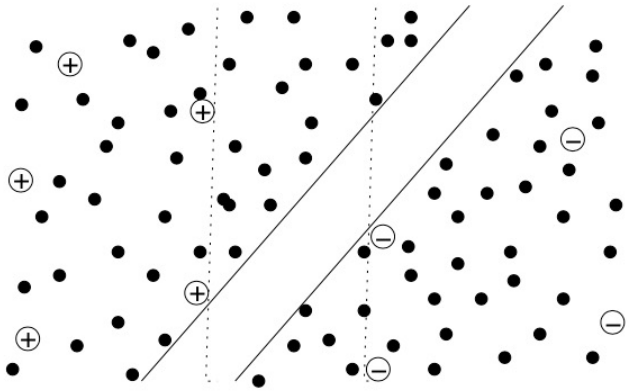
Jizong Peng, Guillermo Estradab, Marco Pedersoli, Christian Desrosiers Deep Co-Training for Semi-Supervised Image Segmentation. European Conference on Computer Vision 2018.

Co-training

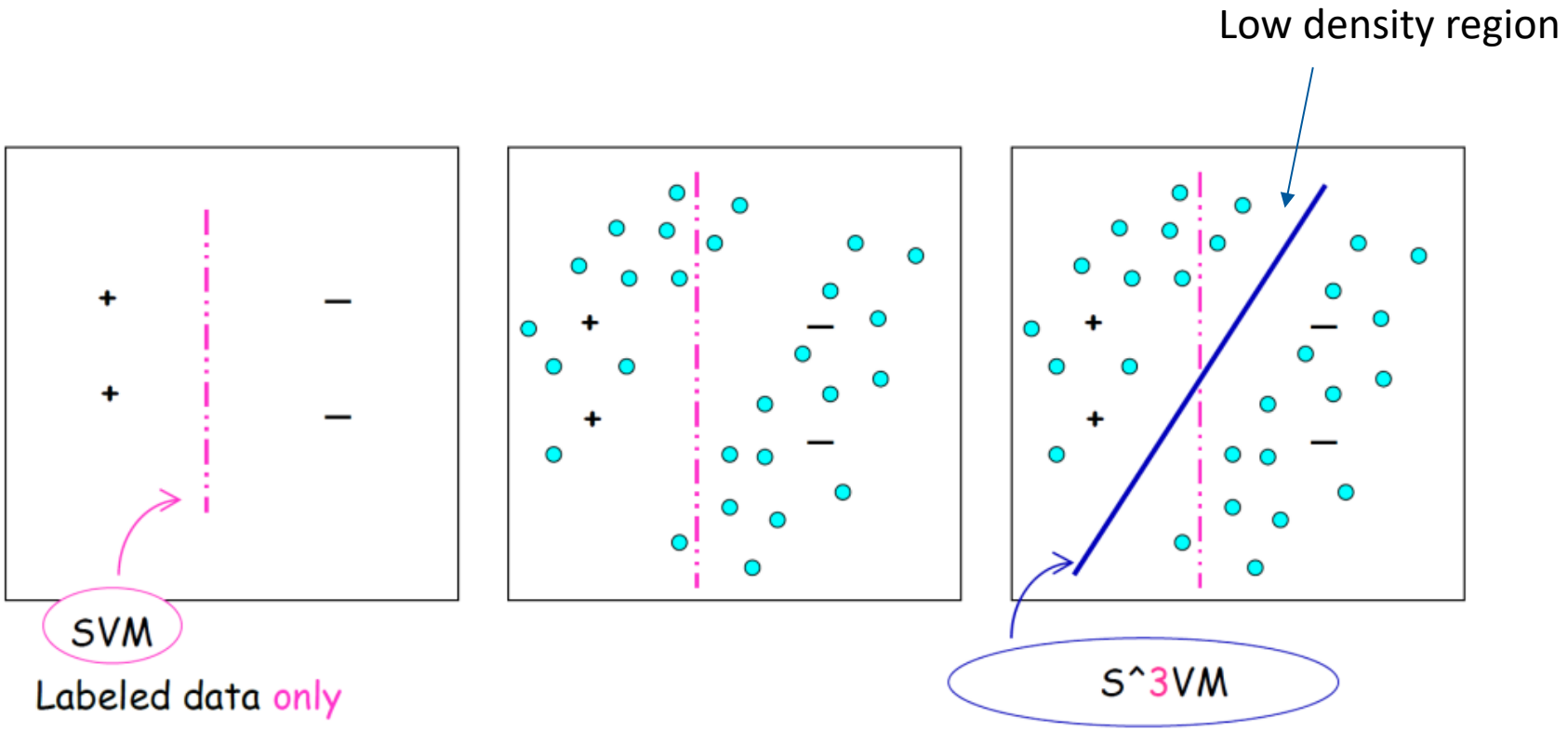
- Advantages:
  - Simple wrapper approach: apply to any model f
  - Less sensitive to prediction mistake than self-training
- Disadvantages
  - Natural split of feature sources can be hard to obtain
  - Models using BOTH features should do better.

S3VM: Semi-supervised Support Vector Machine

- Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)
- Maximizes “unlabeled data margin”
- Assumption: we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin with respect to labelled and unlabelled data. (L+U)



S3SVM: Semi-supervised Support Vector Machine



SVM Recap

$$\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i}\xi_i$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

How to understand this term?

SVM Recap

$$\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i}\xi_i$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$

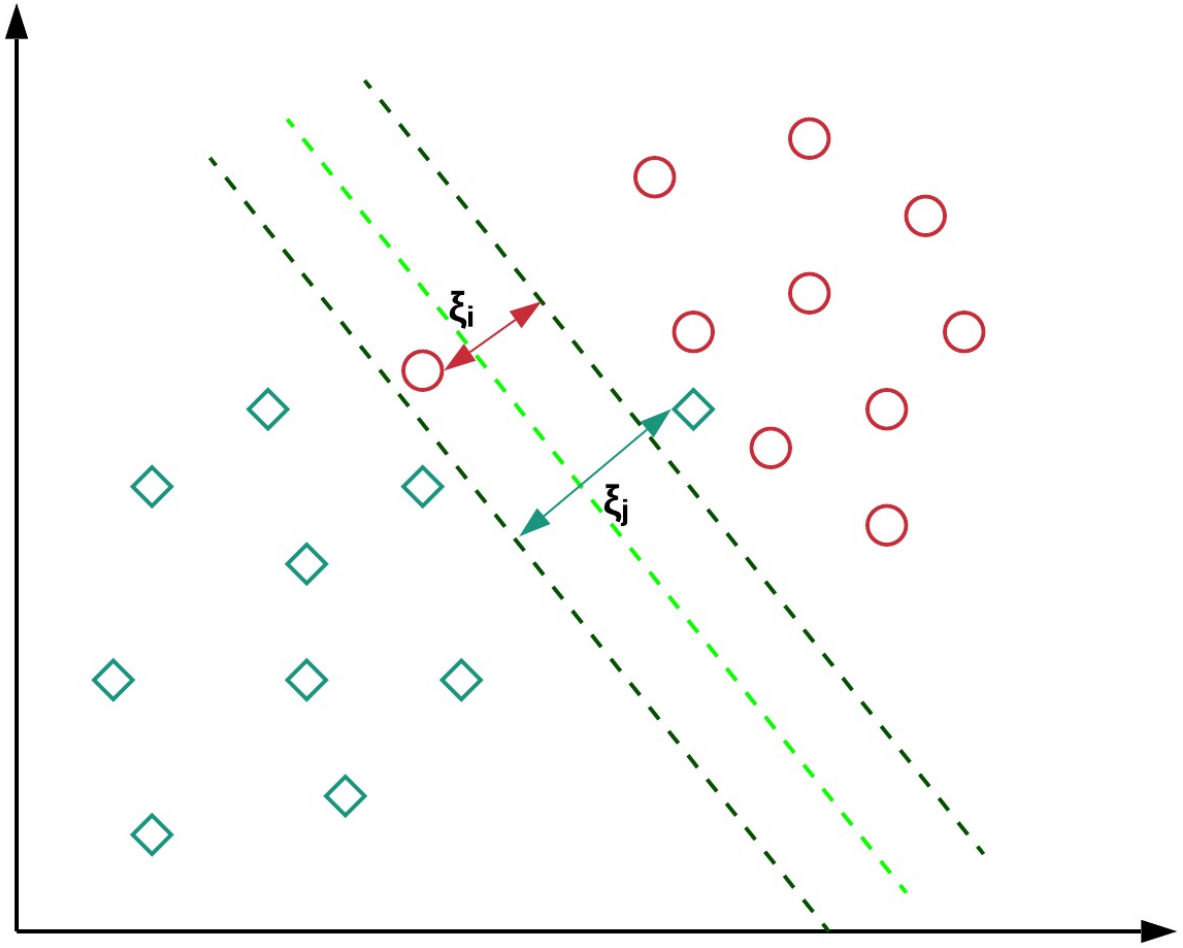
$$\xi_i \geq 0$$

How to understand this term?

It measures how much it violates the hard margin constraints. Recall that in hard margin case, we expect that

$$s.t. \quad y_i(w^T x_i + b) \geq 1$$

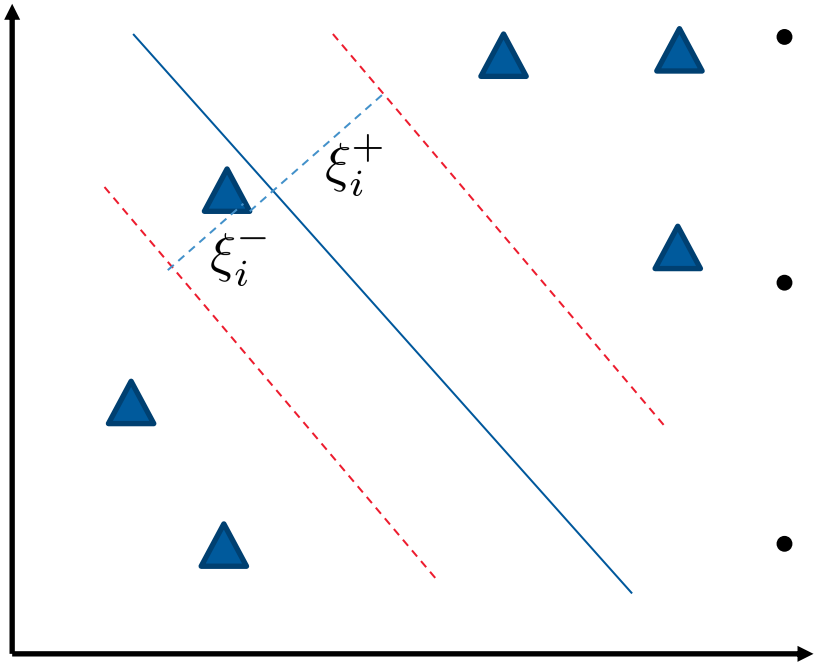
Geometrically speaking



S3VM: Encourage w going through low density regions

- S3SVM will repurpose the interpretation of ξ_i to enforce its key idea: encourage w going through low density regions

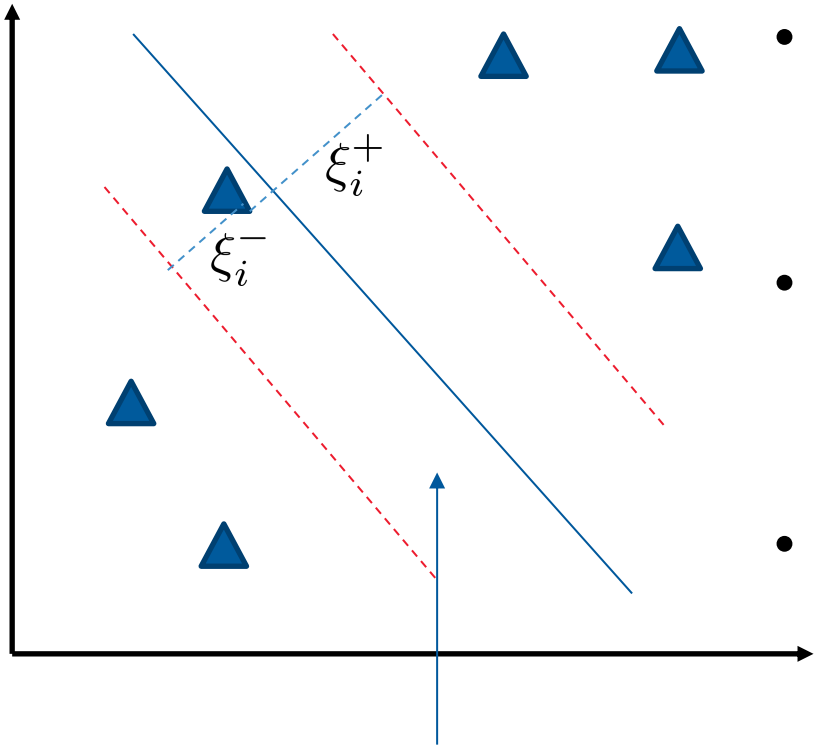
Solution



- We want separator goes through low density regions of the space/large margin.
- Assume each unlabelled sample being class 1 or -1. Calculate loss ξ_i^+ and ξ_i^-
- We want to minimize

$$\min\{\xi_i^+, \xi_i^-\}$$

Solution



- We want separator goes through low density regions of the space/large margin.
- Assume each unlabelled sample being class 1 or -1. Calculate loss ξ_i^+ and ξ_i^-
- We want to minimize

$$\min\{\xi_i^+, \xi_i^-\}$$

According to this scheme, any samples falling between those two red dash lines will be penalized, that is, incurring nonzero loss. Thus the optimization process will seek to place the decision boundary to the region which leads to minimal loss, equivalently, low density region.

Formulation of S3VM: [optional]

$$\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i\in X_l}\xi_i + \mu\sum_{j\in X_u}\min\{\xi_j^+, \xi_j^-\}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i$$

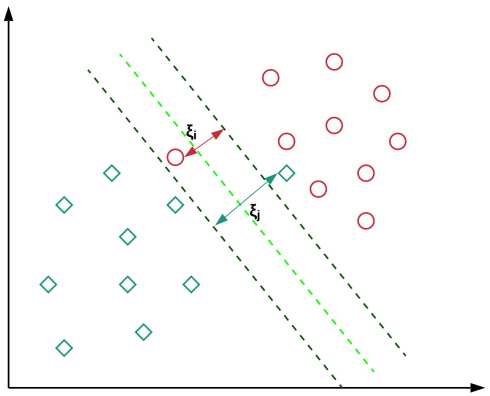
$$(w^T x_j + b) \geq 1 - \xi_j^+ \quad \longleftarrow \text{ Assume y = 1}$$

$$-(w^T x_j + b) \geq 1 - \xi_j^- \quad \longleftarrow \text{ Assume y = -1}$$

$$\xi_i \geq 0$$

$$\xi_j^+ \geq 0$$

$$\xi_j^- \geq 0$$



S3VM: Semi-supervised Support Vector Machine

- It is not a convex problem. More advanced optimization algorithms are needed.
- The key idea and formulation applies to other semi-supervised learning approach

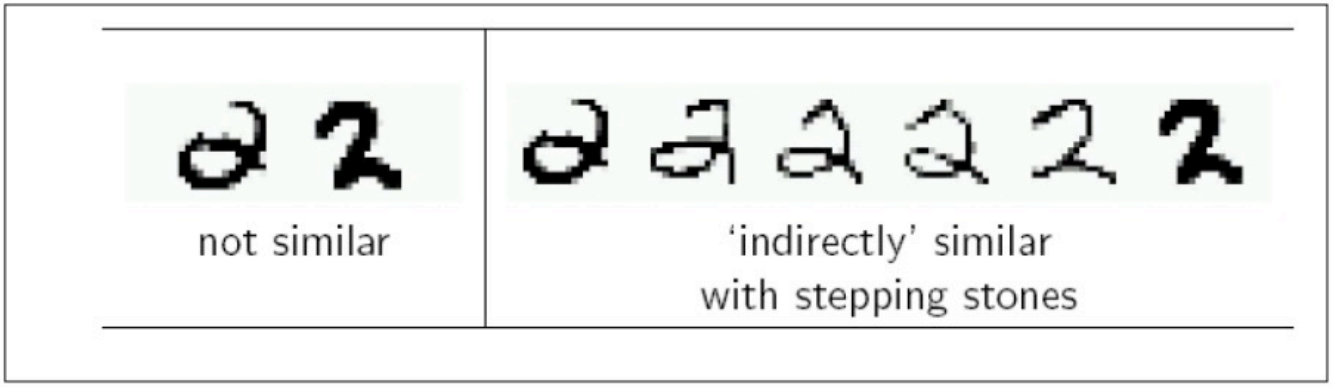
S3VM: Semi-supervised Support Vector Machine

- Advantages:
  - Applicable to wherever SVMs are applicable
  - Clear mathematical formulation

- Disadvantage:
  - Optimization can be difficult
  - Can be trapped into bad local minima

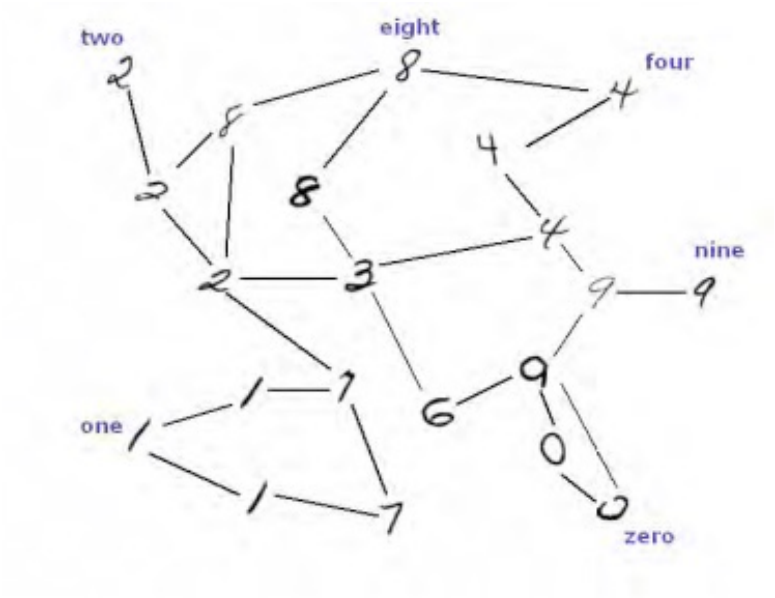
Graph-based semi-supervised learning

- Assumption: we believe that very similar examples probably have the same label.
- We can use unlabelled data as "stepping stones" to propagate the similarity and label



Graph-based semi-supervised learning

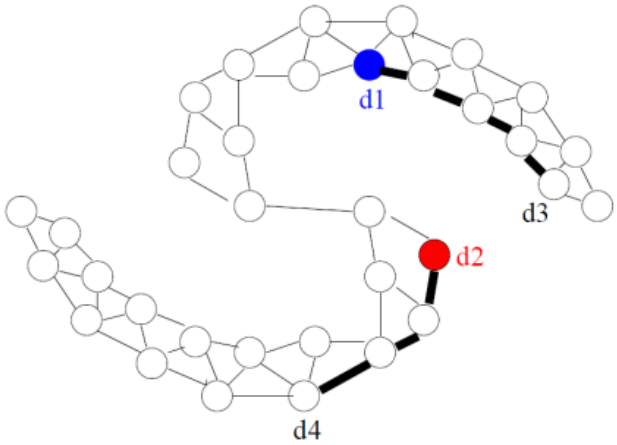
- Idea: Construct a graph on the labeled and unlabeled data. Instances connected by heavy edge tend to have the same label.
- Unlabelled data can help "glue" the objects of the same class together.



Graph-based semi-supervised learning

The Graph

- Nodes: $X_l \cup X_u$
- Edges: similarity weights computed from features, e.g.,
  - k-nearest-neighbor graph, unweighted (0, 1 weights)
  - fully connected graph, weight decays with distance $w = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$
- Want: implied similarity via all paths



Graph-based semi-supervised learning

- Key idea: if two samples are neighbours, the prediction values of them should be similar

$$\min_f \|f(x) - f(x')\| \text{ if } x \text{ and } x' \text{ are neighbours}$$

Graph-based semi-supervised learning

- Formulation:

$$\min_{w} \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{|\mathcal{D}_l|}\sum_{(\mathbf{x},y)\in\mathcal{D}_l} \max(0, 1 - y\langle w, \mathbf{x}\rangle) + \\ + \frac{\gamma}{|\mathcal{D}_u|^2}\sum_{\mathbf{x}\in\mathcal{D}_u}\sum_{\mathbf{x}'\in\mathcal{D}_l\cup\mathcal{D}_u} s(\mathbf{x},\mathbf{x}')(\langle w, \mathbf{x}\rangle - \langle w, \mathbf{x}'\rangle)^2$$

s(x, x') represents the the strength of the connection (or "similarity") on the graph. Different values of s(x, x') lead to different regularization strength.

Graph-based semi-supervised learning

- Formulation:

$$\min_{w} \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{|\mathcal{D}_l|}\sum_{(\mathbf{x},y)\in\mathcal{D}_l} \max(0, 1 - y\langle w, \mathbf{x}\rangle) + \\ + \frac{\gamma}{|\mathcal{D}_u|^2}\sum_{\mathbf{x}\in\mathcal{D}_u}\sum_{\mathbf{x}'\in\mathcal{D}_l\cup\mathcal{D}_u} s(\mathbf{x},\mathbf{x}')(\langle w, \mathbf{x}\rangle - \langle w, \mathbf{x}'\rangle)^2$$

First two terms: same as SVM, you can replace them with any linear classifier
Last term: encouraging similar samples having similar prediction

Graph-based semi-supervised learning

- Advantages:
  - Good performance if the graph fits the task
  - Clear mathematical formulation

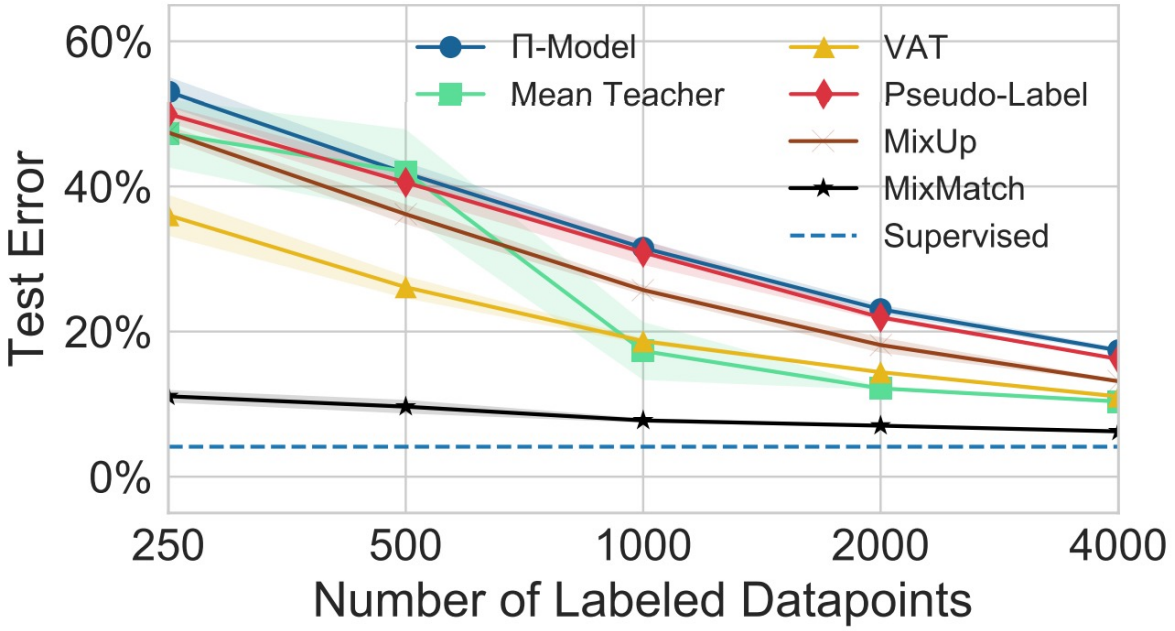
- Disadvantage:
  - Performance is bad if the graph is bad
  - Not scalable to large dataset

Deep semi-supervised learning

- What has changed in the context of deep learning
  - Training methods: from convex optimization to SGD
  - Scale of data: Large scale
  - The structure of predictive model: simple vs. complex, layerwise
  - Feature representation learning with deep learning

Deep Semi-supervised Learning

- A very active research field
- Again, many methods
- Current research progress



Consistency-based deep semi-supervised learning

- Require the network to be less sensitive to the input perturbation
  - Supervision signal not relying on the labels

$$\min_f \|f(x) - f(x')\| \; x' \text{ is the perturbed version of } x$$

- How to make a perturbation
  - Add noise on the input (e.g., images)
  - Data augmentation: For images, shifting, mirroring, color jittering etc.
  - Data augmentation: For text, back translation

[1] Samuli Laine, Timo Aila. Temporal Ensembling for Semi-Supervised Learning. ICLR 2017
[2] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, Quoc V. Le. Unsupervised Data Augmentation for Consistency Training. Arxiv 29 Apr 2019.

Consistency-based deep semi-supervised learning

- Loss function

$$\min_f \sum_{i \in X_l} \mathcal{L}(f(x_i), y_i) + \mu \sum_{j \in X_l \cup X_u} \|f(x_j) - f(x'_j)\|$$

- Problem:
  - At the beginning of training, f may not generate meaningful outputs. Enforcing consistency may result in trivial solutions
  - Solution: weight rump up

$$\min_f \sum_{i \in X_l} \mathcal{L}(f(x_i), y_i) + w(t) \sum_{j \in X_l \cup X_u} \|f(x_j) - f(x'_j)\|$$

Conclusion

- Semi-supervised learning is an area of increasing importance in Machine Learning.
- Automatic methods of collecting data make it more important than ever to develop methods to make use of unlabelled data