Introduction to Semi-Supervised Learning

Matthew King-Roskamp[1]

[1]Department of Mathematics
McGill University

An Overview

1. What is Semi-supervised Learning?
   - And what are all these sub-categories?
   - Fundamental Assumptions
2. Intrinsically semi-supervised setting
   - Methodologies: an inexhaustive list.
   - S_3VM's
   - Pertubation methods: Neural Networks with unsupervised losses.

The goal of this talk is to clearly state the problem and assumptions of semi-supervised learning, and show how one generalizes supervised learning techniques (and losses) to this new setting.

Follow treatment of Van Engelen [VEH20], and textbooks of Chapelle, Zhu [CSZ09, ZG09].

Problem statement

- Input space $\mathcal{X}$, label space $\mathcal{Y}$. data follows some distribution $p(x), p(x, y)$.

  Given some set of labelled points $X_L = \{(x_i, y_i)\}_{i \in L}$, and some set of unlabeled point $X_U = \{x_i\}_{i \in U}$.
- Why this setting?: Particularly useful in the Label-sparse case, where training a reliable learner on labelled data only may be difficult.
- Fundamentally, we wish to use unlabelled data to inform a better "prediction". But what is our goal?

Problem statement

- Input space $\mathcal{X}$, label space $\mathcal{Y}$. data follows some distribution $p(x), p(x, y)$.

  Given some set of labelled points $X_L = \{(x_i, y_i)\}_{i \in L}$, and some set of unlabeled point $X_U = \{x_i\}_{i \in U}$.

- Why this setting?: Particularly useful in the Label-sparse case, where training a reliable learner on labelled data only may be difficult.

- Fundamentally, we wish to use unlabelled data to inform a better "prediction". But what is our goal?

Problem statement

- Input space $\mathcal{X}$, label space $\mathcal{Y}$. data follows some distribution $p(x), p(x, y)$.

  Given some set of labelled points $X_L = \{(x_i, y_i)\}_{i \in L}$, and some set of unlabeled point $X_U = \{x_i\}_{i \in U}$.
- Why this setting?: Particularly useful in the Label-sparse case, where training a reliable learner on labelled data only may be difficult.
- Fundamentally, we wish to use unlabelled data to inform a better "prediction". But what is our goal?

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \rightarrow \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

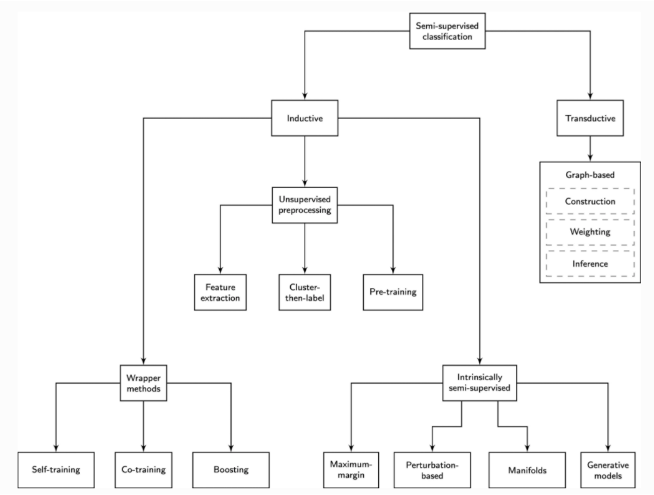
- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
  - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
  - Unsupervised preproccessing of all the data
  - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

Goals (and lots of terminology):

- Do we want to label only those points X_U? This is transductive learning.
- Do we want to find a model $f : \mathcal{X} \to \mathcal{Y}$? This is inductive learning. How this model depends on the unlabelled data, can of course differ!
    - Pseudo-label the unlabelled data, and use supervised methods: Wrapper
    - Unsupervised preproccessing of all the data
    - Use the unlabelled point directly in the objective: Intrinsically semi supervised
- These methods are all discriminative, they learn a function to classify points.
- If one instead tries to model the data generation process, this is generative modeling.

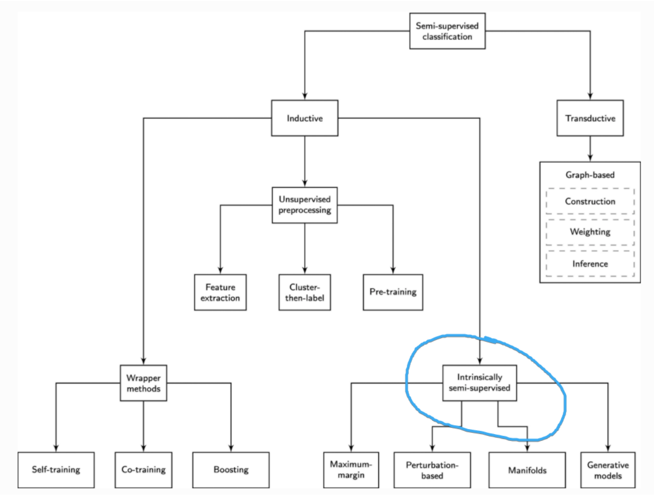
What are *we* interested in?

Where does contrastive learning fit? Figure of [VEH20].



What are *we* interested in?

Where does contrastive learning fit? Figure of [VEH20].



Fundamental assumptions

- For this problem to be sensible, we need our unlabelled data to contain some information about our labels.
- That is $p(x)$ **must** contain some information about $p(y|x)$. If this is not met, it is "inherently impossible to improve predictions based on the additional data" [ZG09].
- So how do we enforce this assumption?

Fundamental assumptions

- For this problem to be sensible, we need our unlabelled data to contain some information about our labels.
- That is $p(x)$ **must** contain some information about $p(y|x)$. If this is not met, it is "inherently impossible to improve predictions based on the additional data" [ZG09].
- So how do we enforce this assumption?

Fundamental assumptions

- For this problem to be sensible, we need our unlabelled data to contain some information about our labels.
- That is $p(x)$ **must** contain some information about $p(y|x)$. If this is not met, it is "inherently impossible to improve predictions based on the additional data" [ZG09].
- So how do we enforce this assumption?

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?

- "If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method." [VEH20]

- These are all fundamentally clustering assumptions.

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?

- “If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method.” [VEH20]

- These are all fundamentally clustering assumptions.

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?
- "If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method." [VEH20]
- These are all fundamentally clustering assumptions.

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?

- “If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method.” [VEH20]

- These are all fundamentally clustering assumptions.

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?

- "If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method." [VEH20]

- These are all fundamentally clustering assumptions.

What sort of form does this assumption take?

There are several ways this can be enforced:

- Points close together have the same label (smoothness)
- The decision boundary is in a region of low density.
- The data lies on some low dimensional manifold.
- Audience: What is the assumption of Contrastive learning?

- “If the data points (both unlabelled and labelled) cannot be meaningfully clustered, it is impossible for a semi-supervised learning method to improve on a supervised learning method.” [VEH20]

- These are all fundamentally clustering assumptions.

Some notes of caution:

If these (or any other model assumptions) fail, semi-supervised approaches can perform worse than fully supervised methods.

This has appeared many times in the literature, either in modelling the underlying distribution, or having samples drawn from a different distribution, etc;
[ZG09, LZ14, OOR+18, SNZ09].

Focus on intrinsically semi-supervised setting

Let us focus on the inductive setting, on intrinsically semi-supervised methods, as this is the most pertinent to us.

Emphasize: this is one leaf, on one node, of a tree of approaches. There is much being left out.

Depending on the enforcement of clustering, or which assumption is made on the on the data, we can further categorize intrinsically semi-supervised methods.

- Maximum margin methods: S_3VM's, gaussian processes, density regularization, pseudo-labeling
- Perturbation methods: Most often Neural networks. Some special types include psuedo-ensembles, Π-models, temporal ensembles, mixup.
- Manifold methods.

Depending on the enforcement of clustering, or which assumption is made on the on the data, we can further categorize intrinsically semi-supervised methods.

- Maximum margin methods: S_3VM's, gaussian processes, density regularization, pseudo-labeling
- Perturbation methods: Most often Neural networks. Some special types include psuedo-ensembles, Π-models, temporal ensembles, mixup.
- Manifold methods.

Depending on the enforcement of clustering, or which assumption is made on the on the data, we can further categorize intrinsically semi-supervised methods.

- Maximum margin methods: S_3VM's, gaussian processes, density regularization, pseudo-labeling
- Perturbation methods: Most often Neural networks. Some special types include psuedo-ensembles, Π-models, temporal ensembles, mixup.
- Manifold methods.

S_3VM's

A toy example to illustrate the natural generalization of supervised methods: SVM's [CSK08].

- The usual SVM problem can be formulated as

$$\begin{aligned} \min_{w,b,\zeta} \quad & \frac{1}{2}\|w\|_2^2 + C \sum_{x_i \in X_L} \zeta_i \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \end{aligned}$$

  for all $i \in L$.
- Question to the audience: what do SVM's try to do?
- Maximize the margin. So we should penalize this for unlabeled points.

S$_3$VM's

A toy example to illustrate the natural generalization of supervised methods: SVM's [CSK08].

- The usual SVM problem can be formulated as

$$\begin{aligned} \min_{w,b,\zeta} \quad & \frac{1}{2}\|w\|_2^2 + C \sum_{x_i \in X_L} \zeta_i \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \end{aligned}$$

  for all $i \in L$.
- Question to the audience: what do SVM's try to do?
- Maximize the margin. So we should penalize this for unlabeled points.

S_3VM's

A toy example to illustrate the natural generalization of supervised methods: SVM's [CSK08].

- The usual SVM problem can be formulated as

$$\begin{aligned} \min_{w,b,\zeta} \quad & \frac{1}{2}\|w\|_2^2 + C \sum_{x_i \in X_L} \zeta_i \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \end{aligned}$$

  for all $i \in L$.
- Question to the audience: what do SVM's try to do?
- Maximize the margin. So we should penalize this for unlabeled points.

S_3VM's

A toy example to illustrate the natural generalization of supervised methods: SVM's [CSK08].

- A semi-supervised SVM penalizes unlabeled points based on their distance to the decision boundary!

$$\min_{w,b,\zeta} \quad \frac{1}{2}\|w\|_2^2 + C_1 \sum_{x_i \in X_L} \zeta_i + C_2 \sum_{x_j \in X_U} \zeta_j$$

$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1 - \zeta_i, i \in L$$

$$|w^T x_j + b| \geq 1 - \zeta_j, j \in U$$

$$\zeta_i \geq 0, \forall i$$

- Question to the audience: what do SVM's try to do?
- Maximize the margin. So we should penalize this for unlabeled points.

Taking this idea to Neural networks.

"The simplicity and efficiency of the backpropagation algorithm for a great variety of loss functions make it attractive to simply add an unsupervised component to a loss function $\mathcal{L}$". [VEH20]

As one might expect, the problem of designing a loss function is problem dependent, and open ended. Some ideas that have been tried with success include:

- Add noise to inputs, and use the network as an encoder/decoder pair and penalize differences [RBH+15]
- Perturb the network itself, and penalize activations of different neurons. [BAP14, LA16]
- Augment inputs with random noise, and penalize their distance. [ZZ09]
- And so on.

Taking this idea to Neural networks.

“The simplicity and efficiency of the backpropagation algorithm for a great variety of loss functions make it attractive to simply add an unsupervised component to a loss function $\mathcal{L}$”. [VEH20]

As one might expect, the problem of designing a loss function is problem dependent, and open ended. Some ideas that have been tried with success include:

- Add noise to inputs, and use the network as an encoder/decoder pair and penalize differences [RBH+15]
- Perturb the network itself, and penalize activations of different neurons. [BAP14, LA16]
- Augment inputs with random noise, and penalize their distance. [ZZ09]
- And so on.

Taking this idea to Neural networks.

“The simplicity and efficiency of the backpropagation algorithm for a great variety of loss functions make it attractive to simply add an unsupervised component to a loss function $\mathcal{L}$”. [VEH20]

As one might expect, the problem of designing a loss function is problem dependent, and open ended. Some ideas that have been tried with success include:

- Add noise to inputs, and use the network as an encoder/decoder pair and penalize differences [RBH+15]
- Perturb the network itself, and penalize activations of different neurons. [BAP14, LA16]
- Augment inputs with random noise, and penalize their distance. [ZZ09]
- And so on.

Taking this idea to Neural networks.

"The simplicity and efficiency of the backpropagation algorithm for a great variety of loss functions make it attractive to simply add an unsupervised component to a loss function $\mathcal{L}$". [VEH20]

As one might expect, the problem of designing a loss function is problem dependent, and open ended. Some ideas that have been tried with success include:

- Add noise to inputs, and use the network as an encoder/decoder pair and penalize differences [RBH$^+$15]
- Perturb the network itself, and penalize activations of different neurons. [BAP14, LA16]
- Augment inputs with random noise, and penalize their distance. [ZZ09]
- And so on.

Takeaways and conclusion

- We now have some terminology for the problems we are interested in.
- The assumptions of semi-supervised learning fundamentally are clustering assumptions.

[BAP14] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in neural information processing systems*, pages 3365–3373, 2014.

[CSK08] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233, 2008.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[LA16] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[LZ14] Yu-Feng Li and Zhi-Hua Zhou. Towards making unlabeled data never hurt. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):175–188, 2014.

[OOR+18] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3235–3246, 2018.

[RBH$^+$15] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[RCVW10] Frédéric Ratle, Gustavo Camps-Valls, and Jason Weston. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2271–2282, 2010.

[SNZ09] Aarti Singh, Robert Nowak, and Jerry Zhu. Unlabeled data: Now it helps, now it doesn't. In *Advances in neural information processing systems*, pages 1513–1520, 2009.

[VEH20] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[ZG09] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[Zhu05] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[ZZ09] Wen Zhang and Quan Zheng. Tsfs: A novel algorithm for single view co-training. In *2009 International Joint Conference on Computational Sciences and Optimization*, volume 1, pages 492–496. IEEE, 2009.