Introduction to Semi-supervised Learning

Pierre-Alexandre Mattei

http://pamattei.github.io – @pamattei
pierre-alexandre.mattei@inria.fr

MSc Data Science

What is semi-supervised learning?

Recap on generative/discriminative models

Semi-supervised generative models

Semi-supervised learning for discriminative models

What is semi-supervised learning?

Recap on generative/discriminative models

Semi-supervised generative models

Semi-supervised learning for discriminative models

Three kinds of supervision

- supervised learning: we have some fully observed labelled data $(x_1, y_1)..., (x_n, y_n)$,

Three kinds of supervision

- supervised learning: we have some fully observed labelled data $(x_1, y_1)..., (x_n, y_n)$,
- unsupervised learning: we have unlabelled data $x_1, ..., x_{n_1}$,

Three kinds of supervision

- supervised learning: we have some fully observed labelled data $(x_1, y_1)..., (x_n, y_n)$,
- unsupervised learning: we have unlabelled data $x_1, ..., x_{n_1}$,
- semi-supervised learning is in-between: we have some unlabelled data $\tilde{x}_1, ..., \tilde{x}_n$, and some labelled data $(x_1, y_1)..., (x_{n_2}, y_{n_2})$. The goal is usually to learn to predict y given x by leveraging both unlabelled and labelled data.

Why is semi-supervised learning important?

- Often, collecting labels is expensive, but plenty of unlabelled data are easily available (e.g. natural images) collecting labels is expensive

Why is semi-supervised learning important?

- Often, collecting labels is expensive, but plenty of unlabelled data are easily available (e.g. natural images) collecting labels is expensive
- Collecting labels may even be forbidden, or impossible in practice...

Why is semi-supervised learning important?

- Often, collecting labels is expensive, but plenty of unlabelled data are easily available (e.g. natural images) collecting labels is expensive
- Collecting labels may even be forbidden, or impossible in practice...
- Often, the data sets that we encounter in class do not have unlabelled data, because they have been "cleaned". This "cleaning" may have destroyed some valuable information, or may have even created some biases in the data.

Ways to do semi-supervised learning (SSL)

SSL is a particular instance of a missing data problem. There are many ways to deal with missing data in machine learning. Today we'll only focus on the ones linked specifically to SSL.

Exercise: any simple ideas on how to do SSL?

Ways to do semi-supervised learning (SSL)

SSL is a particular instance of a missing data problem. There are many ways to deal with missing data in machine learning. Today we'll only focus on the ones linked specifically to SSL.

Exercise: any simple ideas on how to do SSL?

A successful idea is to take a good supervised model, and adapt it to handle missing labels. As we saw before, there are two schools of supervised learning:

- the generative school, which models both features and labels
- the discriminative school, which models the distribution of the labels given the features

What is semi-supervised learning?

Recap on generative/discriminative models

Semi-supervised generative models

Semi-supervised learning for discriminative models

Probabilistic models for regression/classification

We have two random variables $x, y \sim p(x, y)$. The label y can be discrete or continuous, depending on whether we're doing classification or regression. We want to learn p to make probabilistic predictions i.e. to compute $p(y|x)$.

Probabilistic models for regression/classification

We have two random variables $x, y \sim p(x, y)$. The label y can be discrete or continuous, depending on whether we're doing classification or regression. We want to learn p to make probabilistic predictions i.e. to compute $p(y|x)$.

How to model $p(x, y)$? How to infer it?

How to model $p(x, y)$? How to infer it?

There are two main approaches for building $p(x, y)$:

How to model $p(x, y)$? How to infer it?

There are two main approaches for building $p(x, y)$:

- The fully generative (or model-based) approach posits a joint distribution $p(x, y)$ (often by specifying both $p(y)$ and $p(x|y)$).

How to model $p(x, y)$? How to infer it?

There are two main approaches for building $p(x, y)$:

- The fully generative (or model-based) approach posits a joint distribution $p(x, y)$ (often by specifying both $p(y)$ and $p(x|y)$).
- The discriminative (or conditional) approach just specifies $p(y|x)$ and completely ignores $p(x)$.

Exercise: give a few examples of these two approaches.

How to model $p(x, y)$? How to infer it?

There are two main approaches for building $p(x, y)$:

- The fully generative (or model-based) approach posits a joint distribution $p(x, y)$ (often by specifying both $p(y)$ and $p(x|y)$).
- The discriminative (or conditional) approach just specifies $p(y|x)$ and completely ignores $p(x)$.

Exercise: give a few examples of these two approaches.

Generative: naive Bayes, linear discriminant analysis, most of the models Charles Bouveyron talked about in his part of the course (cf. his book with G. Celeux, B. Murphy et A. Raftery)

How to model $p(x, y)$? How to infer it?

There are two main approaches for building $p(x, y)$:

- The fully generative (or model-based) approach posits a joint distribution $p(x, y)$ (often by specifying both $p(y)$ and $p(x|y)$).
- The discriminative (or conditional) approach just specifies $p(y|x)$ and completely ignores $p(x)$.

Exercise: give a few examples of these two approaches.

Generative: naive Bayes, linear discriminant analysis, most of the models Charles Bouveyron talked about in his part of the course (cf. his book with G. Celeux, B. Murphy et A. Raftery)

Discriminative: Neural nets, logistic regression

Discriminative vs generative: the semi supervised case

For SSL, the generative approach is the most natural. Indeed, $p(x, y)$ induces a marginal distribution

$$p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$$

on the unlabelled data x. Maximum likelihood is then natural to set up.

Discriminative vs generative: the semi supervised case

For SSL, the generative approach is the most natural. Indeed, $p(x, y)$ induces a marginal distribution

$$p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$$

on the unlabelled data x. Maximum likelihood is then natural to set up.

Regarding the discriminative approach, things get a bit trickier: we often need to rely on heuristics. Most of the time, we maximise a penalised version of the fully supervised loss, where the penalty depends on the unlabelled data.

A simple generative model, and its marginal distribution

Let us take a simple generative model for binary classification, for example one with Gaussian classes:

$$y \sim p(y|\pi) = \mathcal{B}(y|\pi), \quad p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y).$$

The parameters are $\pi \in [0, 1]$ (class proportions), and the means and covariances of the classes $\mu_0, \mu_1, \Sigma_0, \Sigma_1$.

A simple generative model, and its marginal distribution

Let us take a simple generative model for binary classification, for example one with Gaussian classes:

$$y \sim p(y|\pi) = \mathcal{B}(y|\pi), \quad p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y).$$

The parameters are $\pi \in [0, 1]$ (class proportions), and the means and covariances of the classes $\mu_0, \mu_1, \Sigma_0, \Sigma_1$.

Question: What is the marginal distribution of some unlabeled data x?

A simple generative model, and its marginal distribution

Let us take a simple generative model for binary classification, for example one with Gaussian classes:

$$y \sim p(y|\pi) = \mathcal{B}(y|\pi), \quad p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y).$$

The parameters are $\pi \in [0, 1]$ (class proportions), and the means and covariances of the classes $\mu_0, \mu_1, \Sigma_0, \Sigma_1$.

Question: What is the marginal distribution of some unlabeled data x?

$$p(x) = \int p(x|y)p(y)dy = \pi\mathcal{N}(x|\mu_1, \Sigma_1) + (1 - \pi)\mathcal{N}(x|\mu_0, \Sigma_0),$$

which is a mixture of two Gaussians.

Writing down the likelihood for SSL

We have some unlabelled data $\tilde{x}_1, ..., \tilde{x}_{n_1}$, and some labelled data $(x_1, y_1)..., (x_{n_2}, y_{n_2})$. The likelihood of all the observed data will therefore be

$$\log p\left(\tilde{x}_1, ..., \tilde{x}_n, (x_1, y_1)..., (x_{n_2}, y_{n_2})\right),$$

which is equal to?

Writing down the likelihood for SSL

We have some unlabelled data $\tilde{x}_1, ..., \tilde{x}_{n_1}$, and some labelled data $(x_1, y_1)..., (x_{n_2}, y_{n_2})$. The likelihood of all the observed data will therefore be

$$\log p\left(\tilde{x}_1, ..., \tilde{x}_n, (x_1, y_1)..., (x_{n_2}, y_{n_2})\right),$$

which is equal to?

$$\sum_{i=1}^{n_1} \log p(\tilde{x}_i) + \sum_{i=1}^{n_2} \log p(y_i, x_i),$$

and, using $p(y_i, x_i) = p(x_i|y_i)p(y_i)$

$$\sum_{i=1}^{n_1} \log\left(\pi \mathcal{N}(\tilde{x}_i|\mu_1, \Sigma_1) + (1-\pi)\mathcal{N}(\tilde{x}_i|\mu_0, \Sigma_0)\right)$$

$$+ \sum_{i=1}^{n_2} \left(\log \mathcal{N}(x_i|\mu_{y_i}, \Sigma_{y_i}) + y_i \log \pi + (1-y_i)\log(1-\pi)\right).$$

The red term can be seen as a regulariser that leverage unlabelled data.

Maximising the likelihood for SSL

So we want to maximise the function of $\pi, \mu_0, \mu_1, \Sigma_0, \Sigma_1$

$$\sum_{i=1}^{n_1} \log\left(\pi \mathcal{N}(\tilde{x}_i|\mu_1, \Sigma_1) + (1-\pi)\mathcal{N}(\tilde{x}_i|\mu_0, \Sigma_0)\right)$$

$$+ \sum_{i=1}^{n_2} \left(\log \mathcal{N}(x_i|\mu_{y_i}, \Sigma_{y_i}) + y_i \log \pi + (1-y_i)\log(1-\pi)\right).$$

Any idea how to do that?

Maximising the likelihood for SSL

So we want to maximise the function of $\pi, \mu_0, \mu_1, \Sigma_0, \Sigma_1$

$$\sum_{i=1}^{n_1} \log\left(\pi \mathcal{N}(\tilde{x}_i|\mu_1, \Sigma_1) + (1-\pi)\mathcal{N}(\tilde{x}_i|\mu_0, \Sigma_0)\right)$$

$$+ \sum_{i=1}^{n_2} \left(\log \mathcal{N}(x_i|\mu_{y_i}, \Sigma_{y_i}) + y_i \log \pi + (1-y_i)\log(1-\pi)\right).$$

Any idea how to do that?

The red term is exactly the log likelihood of a Gaussian mixture with 2 components. Using the same recipe Charles Bouveyron taught you, it is possible to derive an expectation-maximisation (EM) algorithm to maximise the likelihood in this SSL case.

Maximising the likelihood for SSL

So we want to maximise the function of $\pi, \mu_0, \mu_1, \Sigma_0, \Sigma_1$

$$\sum_{i=1}^{n_1} \log\left(\pi \mathcal{N}(\tilde{x}_i|\mu_1, \Sigma_1) + (1-\pi)\mathcal{N}(\tilde{x}_i|\mu_0, \Sigma_0)\right)$$

$$+ \sum_{i=1}^{n_2} \left(\log \mathcal{N}(x_i|\mu_{y_i}, \Sigma_{y_i}) + y_i \log \pi + (1-y_i)\log(1-\pi)\right).$$

Any idea how to do that?

The red term is exactly the log likelihood of a Gaussian mixture with 2 components. Using the same recipe Charles Bouveyron taught you, it is possible to derive an expectation-maximisation (EM) algorithm to maximise the likelihood in this SSL case.
We could also use gradient-based optimisation.

More on generative SSL

- For more on this, and in particular the EM algorithm and R packages, see Chapter 5 of Charle Bouveyron's book *Model-based clustering and classification for data science* (with G. Celeux, B. Murphy & A. Raftery).

More on generative SSL

- For more on this, and in particular the EM algorithm and R packages, see Chapter 5 of Charle Bouveyron's book *Model-based clustering and classification for data science* (with G. Celeux, B. Murphy & A. Raftery).
- Of course, in many cases the Gaussian assumption is not very good. In that case more complex models can be used, like mixtures of Gaussians, variational autoencoders...

More on generative SSL

- For more on this, and in particular the EM algorithm and R packages, see Chapter 5 of Charle Bouveyron's book *Model-based clustering and classification for data science* (with G. Celeux, B. Murphy & A. Raftery).
- Of course, in many cases the Gaussian assumption is not very good. In that case more complex models can be used, like mixtures of Gaussians, variational autoencoders...
- Again, the objective function can be seen as the likelihood of the labelled data with a penalty that depends on the unlabelled data. This general idea will inspire us to extend SSL to discriminative models.

What is semi-supervised learning?

Recap on generative/discriminative models

Semi-supervised generative models

Semi-supervised learning for discriminative models

Discriminative model for semi-supervised learning

We have a discriminative model

$$p(y|x) = \mathcal{B}(y|\pi(x)),$$

where $\pi(x)$ is, for example, a neural net. The general SSL recipe will be to maximise regularised versions of the likelihood:

$$\sum_{i=1}^{n_2} \log p(y_i|x_i) + \lambda \text{Reg}(\text{both labelled and unlabelled data}).$$

Discriminative model for semi-supervised learning

We have a discriminative model

$$p(y|x) = \mathcal{B}(y|\pi(x)),$$

where $\pi(x)$ is, for example, a neural net. The general SSL recipe will be to maximise regularised versions of the likelihood:

$$\sum_{i=1}^{n_2} \log p(y_i|x_i) + \lambda \text{Reg(both labelled and unlabelled data)}.$$

Generative SSLs are a particular case of this with

$$\text{Reg(both labelled and unlabelled data)} = \log p(\tilde{x}_1) + ... + \log p(\tilde{x}_{n_1}).$$

Even in the generative case, it can be worth tuning the λ! In particular, $\lambda < 1$ is interesting because it gives more importance to the labelled data.

Two examples of discriminative SSL

- pseudo-labels: the regulariser is composed of a sum of $\log p(\hat{y}(\tilde{x})|\tilde{x})$ where $\hat{y}(\tilde{x})$ is the prediction given by the model. To make this work, one has to rely on a few tricks. For example, a good idea is to keep only confident predictions in the sum.

Two examples of discriminative SSL

- pseudo-labels: the regulariser is composed of a sum of $\log p(\hat{y}(\tilde{x})|\tilde{x})$ where $\hat{y}(\tilde{x})$ is the prediction given by the model. To make this work, one has to rely on a few tricks. For example, a good idea is to keep only confident predictions in the sum.

- entropy regularisation is also based on the idea that we would want the model to give confident predictions on the unlabelled data. The regulariser is simply

$$\sum_{i=1}^{n_1} \text{Entropy}(p(y|\tilde{x}_i)) = -\sum_{i=1}^{n_1} ((1-p)\log(1-p) + p\log p)\,.$$

For more details, see e.g. the paper by Oliver et al. (NeurIPS 2018), *Realistic Evaluation of Deep Semi-Supervised Learning Algorithms*.