SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore – 35.

DEPARTMENT OF BIOMEDICAL ENGINEERING

UNIT 2

SEMI SUPERVISED LEARNING

Semi-supervised learning is a branch of machine learning that combines supervised and unsupervised learning by using both labeled and unlabeled data to train artificial intelligence (AI) models for classification and regression tasks.

Though semi-supervised learning is generally employed for the same use cases in which one might otherwise use supervised learning methods, it's distinguished by various techniques that incorporate unlabeled data into model training, in addition to the labeled data required for conventional supervised learning.

Semi-supervised learning methods are especially relevant in situations where obtaining a sufficient amount of labeled data is prohibitively difficult or expensive, but large amounts of unlabeled data are relatively easy to acquire. In such scenarios, neither fully supervised nor unsupervised learning methods will provide adequate solutions.

Labeled data and machine learning

Training AI models for prediction tasks like classification or regression typically requires *labeled data*: annotated data points that provide necessary context and demonstrate the correct predictions (output) for each sample input. During training, a *loss function* measures the difference (loss) between the model's predictions for a given input and the "ground truth" provided by that input's label. Models *learn* from these labeled examples by using techniques like gradient descent that update model weights to minimize loss. Because this machine learning process actively involves humans, it is called "supervised" learning.

Properly labeling data becomes increasingly labor-intensive for complex AI tasks. For example, to train an image classification model to differentiate between cars and motorcycles, hundreds (if not thousands) of training images must be labeled "car" or "motorcycle"; for a more detailed computer vision task, like object detection, humans must not only annotate the object(s) each image contains, but where each object is located; for even more detailed tasks, like image segmentation, data labels must annotate *specific pixel-by-pixel boundaries* of different image segments for each image.

Labeling data can thus be particularly tedious for certain use cases. In more specialized machine learning use cases, like drug discovery, genetic sequencing or protein classification, data annotation is not only extremely time-consuming, but also requires very specific domain expertise.

Semi-supervised learning offers a way to extract maximum benefit from a scarce amount of labeled data while also making use of relatively abundant unlabeled data.

Semi-supervised learning vs. supervised learning vs. unsupervised learning

Semi-supervised learning can be thought of as a hybrid of or middle ground between supervised learning and unsupervised learning.

Semi-supervised learning vs supervised learning

The primary distinction between semi- and fully supervised machine learning is that the latter can only be trained using fully labeled datasets, whereas the former uses both labeled and unlabeled data samples in the training process. Semi-supervised learning techniques modify or supplement a supervised algorithm—called the “base learner,” in this context—to incorporate information from unlabeled examples. Labeled data points are used to ground the base learner’s predictions and add structure (like how many classes exist and the basic characteristics of each) to the learning problem.

The goal in training any classification model is for it to learn an accurate *decision boundary*: a line—or, for data with more than two dimensions, a “surface” or hyperplane—separates data points of one classification category from data points belonging to a different classification category. Though a fully supervised classification model can technically learn *a* decision boundary using only a few labeled data points, it might not generalize well to real-world examples, making the model's predictions unreliable.

The classic “half-moons” dataset visualizes the shortcomings of supervised models relying on too few labeled data points. Though the “correct” decision boundary would separate each of the two half-moons, a supervised learning model is likely to overfit the few labeled data points available. The unlabeled data points clearly convey helpful context, but a traditional supervised algorithm cannot process unlabeled data.

Using only the very limited labeled data points available, a supervised model may learn a decision boundary that will generalize poorly and be prone to misclassifying new examples.

Semi-supervised learning vs unsupervised learning

Unlike semi-supervised (and fully supervised) learning, unsupervised learning algorithms use neither labeled data nor loss functions. Unsupervised learning eschews any “ground truth” context against which model accuracy can be measured and optimized.

An increasingly common semi-supervised approach, particularly for large language models, is to “pre-train” models via unsupervised tasks that require the model to learn meaningful representations of unlabeled data sets. When such tasks involve a “ground truth” and loss function (*without* manual data annotation), they’re called self-supervised learning. After subsequent “supervised fine tuning” on a small amount of labeled data, pre-trained models can often achieve performance comparable to fully supervised models.

While unsupervised learning methods can be useful in many scenarios, that lack of context can make them ill-suited to classification on their own. Take, for example, how a typical clustering algorithm—grouping data points into a pre-determined number of clusters based on their proximity to one another—would treat the half-moon dataset.

A typical unsupervised algorithm, k-means clustering, might incorrectly group data points together based only on their relative closeness to "average" datapoints (centroids).

Semi-supervised learning vs self-supervised learning

Both semi- and self-supervised learning aim to circumvent the need for large amounts of labeled data—but whereas semi-supervised learning involves *some* labeled data, self-supervised learning methods like autoencoders are truly unsupervised.

While supervised (and semi-supervised) learning requires an external "ground truth," in the form of labeled data, self-supervised learning tasks derive the ground truth from the underlying structure of unlabeled samples. Many self-supervised tasks are not useful unto themselves: their utility lies in teaching models data representations useful for the purposes of subsequent "downstream tasks." As such, they are often called "pretext tasks."

When combined with supervised downstream tasks, self-supervised pretext tasks thus comprise part of a semi-supervised learning process: a learning method using both labeled and unlabeled data for model training.
How does semi-supervised learning work?

Semi-supervised learning relies on certain assumptions about the unlabeled data used to train the model and the way data points from different classes relate to one another.

A necessary condition of semi-supervised learning (SSL) is that the unlabeled examples used in model training must be relevant to the task the model is being trained to perform. In more formal terms, SSL requires that the distribution $p(x)$ of the input data must contain information about the posterior distribution $p(y|x)$—that is, the conditional probability of a given data point (x) belonging to a certain class (y). So, for example, if one is using unlabeled data to help train an image classifier to differentiate between pictures of cats and pictures of dogs, the training dataset should contain images of both cats and dogs—and images of horses and motorcycles will not be helpful.

Accordantly, while a 2018 study of semi-supervised learning algorithms found that "increasing the amount of unlabeled data tends to improve the performance of SSL techniques," it also found that "adding unlabeled data from a mismatched set of classes can actually *hurt* performance compared to not using any unlabeled data at all." [1]

The basic condition of $p(x)$ having a meaningful relationship to $p(x|y)$ gives rise to multiple **assumptions** about the nature of that relationship. These assumptions are the driving force behind most, if not all, SSL methods: generally speaking, any semi-supervised learning algorithm relies on one or more of the following assumptions being explicitly or implicitly satisfied.

Cluster assumption

The *cluster assumption* states that data points belonging to the same *cluster*–a set of data points more similar to each other than they are to other available data points–will also belong to the same class.

Though sometimes considered to be its own independent assumption, the clustering assumption has also been described by van Engelen and Hoos as “a generalization of the other assumptions."[2] In this view, the determination of data point clusters depends on which notion of similarity is being used: the smoothness assumption, low-density assumption and manifold assumption each simply leverage a different definition of what comprises a “similar” data point.

Smoothness assumption

The *smoothness assumptions* states that if two data points, *x* and *x'*, are close to each other in the input space—the set of all possible values for *x*–then their labels, *y* and *y'*, should be the same.

This assumption, also known as the *continuity assumption*, is common to most supervised learning: for example, classifiers learn a meaningful approximation (or “representation”) of each relevant class during training; once trained, they determine the classification of new data points via which representation they most closely resemble.

In the context of SSL, the smoothness assumption has the added benefit of being applied *transitively* to unlabeled data. Consider a scenario involving three data points:

- a labeled data point, $\boldsymbol{x_1}$
- an unlabeled data point, $\boldsymbol{x_2}$, that’s close to $\boldsymbol{x_1}$
- another unlabeled data point, $\boldsymbol{x_3}$, that’s close to $\boldsymbol{x_2}$ but not close to $\boldsymbol{x_1}$

The smoothness assumption tells us that x_2 should have the same label as x_1. It also tells us that x_3 should have the same label as x_2. Therefore, we can assume that *all three* data points have the same label, because x_1’s label is transitively propagated to x_3 because of x_3’s proximity to x_2.

Low-density assumption

The *low-density assumption* states that the decision boundary between classes should not pass through high-density regions. Put another way, the decision boundary should lie in an area that contains few data points.

The low-density assumption could thus be thought of as an extension of the cluster assumption (in that a high-density cluster of data points represents a class, rather than the boundary between classes) and the smoothness assumption (in that if multiple data points are near each other, they should share a label, and thus fall on the same side of the decision boundary).

This diagram illustrates how the smoothness and low-density assumptions can inform a far more intuitive decision boundary than would be possible with supervised methods that can only consider the (very few) labeled data points.

Source: van Engelen, et al (2018)

Manifold assumption

The *manifold assumption* states that the higher-dimensional input space comprises multiple lower dimensional manifolds on which all data points lie, and that data points on the same manifold share the same label.

For an intuitive example, consider a piece of paper crumpled up into a ball. The location of any points on the spherical surface can only mapped with three-dimensional *x,y,z* coordinates. But if that crumpled up ball is now flattened back into a sheet of paper, those same points can now be mapped with two-dimensional *x,y* coordinates. This is called *dimensionality reduction*, and it can be achieved mathematically using methods like autoencoders or convolutions.

In machine learning, dimensions correspond not to the familiar *physical* dimensions, but to each attribute or feature of data. For example, in machine learning, a small RGB image measuring 32x32 pixels has *3,072* dimensions: 1,024 pixels, each of which has three values (for red, green and blue). Comparing data points with so many dimensions is challenging, both because of the complexity and computational resources required and because most of that high-dimensional space does not contain information meaningful to the task at hand.

The manifold assumption holds that when a model learns the proper dimensionality reduction function to discard irrelevant information, disparate data points converge to a more meaningful representation for which the other SSL assumptions are more reliable.

Mapping the data points to a lower-dimensional manifold can provide a more accurate decision boundary, which can then be translated back to higher-dimensional space. (source: van Engelen, et al, 2018)

Reference:

https://www.ibm.com/topics/semi-supervised-learning#:~:text=Semi%2Dsupervised%20learning%20is%20a,for%20classification%20and%20regression%20tasks.