Preface

During the last years, semi-supervised learning has emerged as an exciting new direction in machine learning reseach. It is closely related to profound issues of how to do inference from data, as witnessed by its overlap with *transductive inference* (the distinctions are yet to be made precise).

At the same time, dealing with the situation where relatively few labeled training points are available, but a large number of unlabeled points are given, it is directly relevant to a multitude of practical problems where is it relatively expensive to produce labeled data, e.g., the automatic classification of web pages. As a field, semi-supervised learning uses a diverse set of tools and illustrates, on a small scale, the sophisticated machinery developed in various branches of machine learning such as kernel methods or Bayesian techniques.

As we work on semi-supervised learning, we have been aware of the lack of an authoritative overview of the existing approaches. In a perfect world, such an overview should help both the practitioner and the researcher who wants to enter this area. A well researched monograph could ideally fill such a gap; however, the field of semi-supervised learning is arguably not yet sufficiently mature for this. Rather than writing a book which would come out in three years, we thus decided instead to provide an up-to-date edited volume, where we invited contributions by many of the leading proponents of the field. To make it more than a mere collection of articles, we have attempted to ensure that the chapters form a coherent whole and use consistent notation. Moreover, we have written a short introduction, a dialogue illustrating some of the ongoing debates in the underlying philosophy of the field, and we have organized and summarized a comprehensive *benchmark* of semi-supervised learning.

provide benchmarks,
on which the
different algorithms
can be compared

Benchmarks are helpful for the practitioner to decide which algorithm should be chosen for a given application. At the same time, they are useful for researchers to choose issues to study and further develop. By evaluating and comparing the performance of many of the presented methods on a set of eight benchmark problems, this book aims at providing guidance in this respect. The problems are designed to reflect and probe the different assumptions that the algorithms build on. All data sets can be downloaded from the book web page, which can be found at `http://www.kyb.tuebingen.mpg.de/ssl-book/`.

Finally, we would like to give thanks to everybody who contributed towards the success of this book project, in particular to Karin Bierig, Sabrina Nielebock, Bob Prior, to all chapter authors, and to the chapter reviewers.

1 Introduction to Semi-Supervised Learning

underlying theme: "know your problem and make the right assumptions"

framing matches current context in many fields well

1.1 Supervised, Unsupervised, and Semi-Supervised Learning

In order to understand the nature of semi-supervised learning, it will be useful first to take a look at supervised and unsupervised learning.

1.1.1 Supervised and Unsupervised Learning

Traditionally, there have been two fundamentally different types of tasks in machine learning.

unsupervised learning

The first one is *unsupervised learning*. Let $X = (x_1, \ldots, x_n)$ be a set of n examples (or points), where $x_i \in \mathcal{X}$ for all $i \in [n] := \{1, \ldots, n\}$. Typically it is assumed that the points are drawn i.i.d. (independently and identically distributed) from a common distribution on $\mathcal{X}$. It is often convenient to define the $(n \times d)$-matrix $\mathbf{X} = (x_i^\top)_{i \in [n]}^\top$ that contains the data points as its rows. The goal of unsupervised learning is to find interesting structure in the data X. It has been argued that the problem of unsupervised learning is fundamentally that of estimating a density which is likely to have generated X. However, there are also weaker forms of unsupervised learning, such as quantile estimation, clustering, outlier detection, and dimensionality reduction.

supervised learning

continuous labels
→ regression
discrete labels
→ classification

The second task is *supervised learning*. The goal is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i). Here, the $y_i \in \mathcal{Y}$ are called the labels or targets of the examples x_i. If the labels are numbers, $\mathbf{y} = (y_i)_{i \in [n]}^\top$ denotes the column vector of labels. Again, a standard requirement is that the pairs (x_i, y_i) are sampled i.i.d. from some distribution which here ranges over $\mathcal{X} \times \mathcal{Y}$. The task is well defined, since a mapping can be evaluated through its predictive performance on test examples. When $\mathcal{Y} = \mathbb{R}$ or $\mathcal{Y} = \mathbb{R}^d$ (or more generally, when the labels are continuous), the task is called regression. Most of this book will focus on classification (there is some work on regression in chapter 23), i.e., the case where y takes values in a finite set (discrete labels). There are two families of algorithms for supervised learning. *Generative* algorithms try to model the class-conditional

generative methods

density $p(x|y)$ by some unsupervised learning procedure.[1] A predictive density can then be inferred by applying Bayes theorem:

$$p(y|x) = \frac{p(x|y)p(y)}{\int_y p(x|y)p(y)dy}. \tag{1.1}$$

discriminative methods

In fact, $p(x|y)p(y) = p(x, y)$ is the joint density of the data, from which pairs (x_i, y_i) could be generated. *Discriminative* algorithms do not try to estimate how the x_i have been generated, but instead concentrate on estimating $p(y|x)$. Some discriminative methods even limit themselves to modeling whether $p(y|x)$ is greater than or less than 0.5; an example of this is the support vector machine (SVM). It has been argued that discriminative models are more directly aligned with the goal of supervised learning and therefore tend to be more efficient in practice. These two frameworks are discussed in more detail in sections 2.2.1 and 2.2.2.

how does this compare to language modeling?,
→ need to read sections

1.1.2 Semi-Supervised Learning

standard setting of SSL

Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information – but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. In this case, the data set $X = (x_i)_{i \in [n]}$ can be divided into two parts: the points $X_l := (x_1, \ldots, x_l)$, for which labels $Y_l := (y_1, \ldots, y_l)$ are provided, and the points $X_u := (x_{l+1}, \ldots, x_{l+u})$, the labels of which are not known. This is "standard" semi-supervised learning as investigated in this book; most chapters will refer to this setting.

SSL with constraints

Other forms of partial supervision are possible. For example, there may be constraints such as "these points have (or do not have) the same target" (cf. Abu-Mostafa, 1995). This more general setting is considered in chapter 5. The different setting corresponds to a different view of semi-supervised learning: In chapter 5, SSL is seen as unsupervised learning guided by constraints. In contrast, most other approaches see SSL as supervised learning with additional information on the distribution of the examples x. The latter interpretation seems to be more in line with most applications, where the goal is the same as in supervised learning: to predict a target value for a given x_i. However, this view does not readily apply if the number and nature of the classes are not known in advance but have to be inferred from the data. In constrast, SSL as unsupervised learning with constraints may still remain applicable in such situations.

two perspectives: from supervised or from unsupervised learning.

transductive learning

inductive learning

A problem related to SSL was introduced by Vapnik already several decades ago: so-called *transductive learning*. In this setting, one is given a (labeled) training set and an (unlabeled) test set. The idea of transduction is to perform predictions only for the test points. This is in contrast to *inductive learning*, where the goal is to

[1] For simplicity, we are assuming that all distributions have densities, and thus we restrict ourselves to dealing with densities.

i.e. without seeking generalization? <—

output a prediction function which is defined on the entire space $\mathcal{X}$. Many methods described in this book will be transductive; in particular, this is rather natural for inference based on graph representations of the data. This issue will be addressed again in section 1.2.4.

1.1.3 A Brief History of Semi-Supervised Learning

self-learning

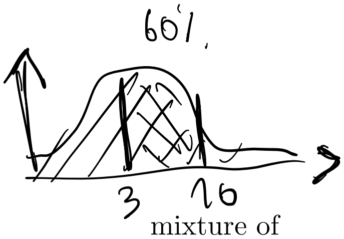
~train on your own predictions

Probably the earliest idea about using unlabeled data in classification is self-learning, which is also known as self-training, self-labeling, or decision-directed learning. This is a wrapper-algorithm that repeatedly uses a supervised learning method. It starts by training on the labeled data only. In each step a part of the unlabeled points is labeled according to the current decision function; then the supervised method is retrained using its own predictions as additional labeled points. This idea has appeared in the literature already for some time (e.g., Scudder (1965); Fralick (1967); Agrawala (1970)).

An unsatisfactory aspect of self-learning is that the effect of the wrapper depends on the supervised method used inside it. If self-learning is used with empirical risk minimization and 1-0-loss, the unlabeled data will have no effect on the solution at all. If instead a margin maximizing method is used, as a result the decision boundary is pushed away from the unlabeled points (cf. chapter 6). In other cases it seems to be unclear what the self-learning is really doing, and which assumption it corresponds to.

transductive inference

Closely related to semi-supervised learning is the concept of transductive inference, or transduction, pioneered by Vapnik (Vapnik and Chervonenkis, 1974; Vapnik and Sterin, 1977). In contrast to inductive inference, no general decision rule is inferred, but only the labels of the unlabeled (or test) points are predicted. An early instance of transduction (albeit without explicitly considering it as a concept) was already proposed by Hartley and Rao (1968). They suggested a combinatorial optimization on the labels of the test points in order to maximize the likelihood of their model.



mixture of Gaussians

It seems that semi-supervised learning really took off in the 1970s when the problem of estimating the Fisher linear discriminant rule with unlabeled data was considered (Hosmer, 1973; McLachlan, 1977; O'Neill, 1978; McLachlan and Ganesalingam, 1982). More precisely, the setting was in the case where each class-conditional density is Gaussian with equal covariance matrix. The likelihood of the model is then maximized using the labeled and unlabeled data with the help of an iterative algorithm such as the expectation-maximization (EM) algorithm (Dempster et al., 1977). Instead of a mixture of Gaussians, the use of a mixture of multinomial distributions estimated with labeled and unlabeled data has been investigated in (Cooper and Freeman, 1970).

assuming that a random sample belongs to class C, we have a Gaussian density function that tells us how probable it is that the sample's value(s) lie in a certain interval

Later, this one component per class setting has been extended to several components per class (Shahshahani and Landgrebe, 1994) and further generalized by Miller and Uyar (1997).

Learning rates in a probably approximately correct (PAC) framework (Valiant, theoretical analysis

1984) have been derived for the semi-supervised learning of a mixture of two Gaussians by Ratsaby and Venkatesh (1995). In the case of an *identifiable* mixture, Castelli and Cover (1995) showed that with an infinite number of unlabeled points, the probability of error has an exponential convergence (w.r.t. the number of labeled examples) to the Bayes risk. Identifiable means that given $P(\mathbf{x})$, the decomposition in $\sum_y P(y)P(\mathbf{x}|y)$ is unique. This seems a relatively strong assumption, but it is satisfied, for instance, by mixtures of Gaussians. Related is the analysis in (Castelli and Cover, 1996) in which the class-conditional densities are known but the class priors are not.

text applications

lots of unlabeled text + some training labels for a certain task

Finally, the interest in semi-supervised learning increased in the 1990s, mostly due to applications in natural language problems and text classification (Yarowsky, 1995; Nigam et al., 1998; Blum and Mitchell, 1998; Collins and Singer, 1999; Joachims, 1999).

Note that, to our knowledge, Merz et al. (1992) were the first to use the term "semi-supervised" for classification with both labeled and unlabeled data. It has in fact been used before, but in a different context than what is developed in this book; see, for instance, (Board and Pitt, 1989).

1.2 When Can Semi-Supervised Learning Work?

important:

semi-supervised learning works, if the labeled data's distribution relevant for problem

A natural question arises: is semi-supervised learning meaningful? More precisely: in comparison with a supervised algorithm that uses only labeled data, can one hope to have a more accurate prediction by taking into account the unlabeled points? As you may have guessed from the size of the book in your hands, in principle the answer is "yes." However, there is an important prerequisite: that the distribution of examples, which the unlabeled data will help elucidate, be relevant for the classification problem.

In a more mathematical formulation, one could say that the knowledge on $p(x)$ that one gains through the unlabeled data has to carry information that is useful in the inference of $p(y|x)$. If this is not the case, semi-supervised learning will not yield an improvement over supervised learning. It might even happen that using the unlabeled data degrades the prediction accuracy by misguiding the inference; this effect is investigated in detail in chapter 4.

One should thus not be too surprised that for semi-supervised learning to work, certain *assumptions* will have to hold. In this context, note that plain supervised learning also has to rely on assumptions. In fact, chapter 22 discusses a way of formalizing assumptions of the kind given below within a PAC-style framework. One of the most popular such assumptions can be formulated as follows.

smoothness assumption

intuitively clear, means representation matters

Smoothness assumption of supervised learning:[2] If two points x_1, x_2 are close, then so should be the corresponding outputs y_1, y_2.

Clearly, without such assumptions, it would never be possible to generalize from a finite training set to a set of possibly infinitely many unseen test cases.

1.2.1 The Semi-Supervised Smoothness Assumption

We now propose a generalization of the smoothness assumption that is useful for semi-supervised learning; we thus call it the "semi-supervised smoothness assumption". While in the supervised case according to our prior beliefs the output varies smoothly with the distance, we now also take into account the density of the inputs. The assumption is that the label function is smoother in high-density regions than in low-density regions:

semi-supervised smoothness assumption

Semi-supervised smoothness assumption: If two points x_1, x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2.

weaker form of smoothness assumed, only holds in clusters

Note that by transitivity, this assumption implies that if two points are linked by a path of high density (e.g., if they belong to the same cluster), then their outputs are likely to be close. If, on the other hand, they are separated by a low-density region, then their outputs need not be close.

Note that the semi-supervised smoothness assumption applies to both regression and classification. In the next section, we will show that in the case of classification, it reduces to assumptions commonly used in SSL. At present, it is less clear how useful the assumption is for regression problems. As an alternative, chapter 23 proposes a way to use unlabeled data for model selection that applies to both regression and classification.

1.2.2 The Cluster Assumption

cluster assumption

Suppose we knew that the points of each class tended to form a cluster. Then the unlabeled data could aid in finding the boundary of each cluster more accurately: one could run a clustering algorithm and use the labeled points to assign a class to each cluster. That is in fact one of the earliest forms of semi-supervised learning (see chapter 2). The underlying, now classical, assumption may be stated as follows:

Cluster assumption: If points are in the same cluster, they are likely to be of the same class.

→ applied to discrete labels / classification

This assumption may be considered reasonable on the basis of the sheer existence

[2] Strictly speaking, this assumption only refers to continuity rather than smoothness; however, the term *smoothness* is commonly used, possibly because in regression estimation y is often modeled in practice as a smooth function of x.

of classes: if there is a densly populated continuum of objects, it may seem unlikely that they were ever distinguished into different classes.

Note that the cluster assumption does not imply that each class forms a single, compact cluster: it only means that, usually, we do not observe objects of two distinct classes in the same cluster.

The cluster assumption can easily be seen as a special case of the above-proposed semi-supervised smoothness assumption, considering that clusters are frequently defined as being sets of points that can be connected by short curves which traverse only high-density regions.

equivalent idea:

low density separation

The cluster assumption can be formulated in an equivalent way:

Low density separation: The decision boundary should lie in a low-density region.

again: two different perspectives, leading to different algorithms

The equivalence is easy to see: A decision boundary in a high-density region would cut a cluster into two different classes; many objects of different classes in the same cluster would require the decision boundary to cut the cluster, i.e., to go through a high-density region.

Although the two formulations are conceptually equivalent, they can inspire different algorithms, as we will argue in section 1.3. The low-density version also gives additional intuition why the assumption is sensible in many real-world problems. Consider digit recognition, for instance, and suppose that one wants to learn how to distinguish a handwritten digit “0” against digit “1”. A sample point taken exactly from the decision boundary will be between a 0 and a 1, most likely a digit looking like a very elongated zero. But the probability that someone wrote this “weird” digit is very small.

1.2.3 The Manifold Assumption

manifold assumption

A different but related assumption that forms the basis of several semi-supervised learning methods is the manifold assumption:

Manifold assumption: The (high-dimensional) data lie (roughly) on a low-dimensional manifold.

curse of dimensionality

How can this be useful? A well-known problem of many statistical methods and learning algorithms is the so-called curse of dimensionality (cf. section 11.6.2). It is related to the fact that volume grows exponentially with the number of dimensions, and an exponentially growing number of examples is required for statistical tasks such as the reliable estimation of densities. This is a problem that directly affects generative approaches that are based on density estimates in input space. A related problem of high dimensions, which may be more severe for discriminative methods, is that pairwise distances tend to become more similar, and thus less expressive.

topological space that locally resembles Euclidean space near each point (e.g maps of the earth)

If the data happen to lie on a low-dimensional manifold, however, then the learning algorithm can essentially operate in a space of corresponding dimension, thus avoiding the curse of dimensionality.

As above, one can argue that algorithms working with manifolds may be seen smoothness assumption can well apply to a lower-dimensional manifold

as approximately implementing the semi-supervised smoothness assumption: such algorithms use the metric of the manifold for computing geodesic distances. If we view the manifold as an approximation of the high-density regions, then it becomes clear that in this case, the semi-supervised smoothness assumption reduces to the standard smoothness assumption of supervised learning, applied on the manifold.

Note that if the manifold is embedded into the high-dimensional input space in a curved fashion (i.e., it is not just a subspace), geodesic distances differ from those in the input space. By ensuring more accurate density estimates and more appropriate distances, the manifold assumption may be useful for classification as well as for regression.

1.2.4 Transduction

As mentioned before, some algorithms naturally operate in a transductive setting. According to the philosophy put forward by Vapnik, high-dimensional estimation problems should attempt to follow the following principle:

Vapnik's principle: When trying to solve some problem, one should not solve a more difficult problem as an intermediate step.

Consider as an example supervised learning, where predictions of labels y corresponding to some objects x are desired. Generative models estimate the density of x as an intermediate step, while discriminative methods directly estimate the labels.

In a similar way, if label predictions are only required for a given test set, transduction can be argued to be more direct than induction: while an inductive method infers a function $f : \mathcal{X} \to \mathcal{Y}$ on the entire space $\mathcal{X}$, and afterward returns the evaluations $f(x_i)$ at the test points, transduction consists of directly estimating the finite set of test labels, i.e., a function $f : X_u \to \mathcal{Y}$ only defined on the test set. Note that transduction (as defined in this book) is not the same as SSL: some semi-supervised algorithms are transductive, but others are inductive.

Now suppose we are given a transductive algorithm which produces a solution superior to an inductive algorithm trained on the same labeled data (but discarding the unlabeled data). Then the performance difference might be due to one of the following two points (or a combination thereof):

1. transduction follows Vapnik's principle more closely than induction does, or
2. the transductive algorithm takes advantage of the unlabeled data in a way similar to semi-supervised learning algorithms.

There is ample evidence for improvements being due to the second of these points. We are presently not aware of empirical results that selectively support the first point. In particular, the evaluation of the benchmark associated with this book (chapter 21) does not seem to suggest a systematic advantage of transductive methods. However, the properties of transduction are still the topic of debate, and chapter 25 tries to present different opinions.

1.3 Classes of Algorithms and Organization of This Book

4 classes of algorithms, summary on this page ff.

Although many methods were not explicitly derived from one of the above assumptions, most algorithms can be seen to correspond to or implement one or more of them. We try to organize the semi-supervised learning methods presented in this book into four classes that roughly correspond to the underlying assumption. Although the classification is not always unique, we hope that this organization makes the book and its contents more accessible to the reader, by providing a guiding scheme.

For the same reason, this book is organized in "parts." There is one part for each class of SSL algorithms and an extra part focusing on generative approaches. Two further parts are devoted to applications and perspectives of SSL. In the following we briefly introduce the ideas covered by each book part.

I

1.3.1 Generative Models

Part I presents history and state of the art of SSL with generative models. Chapter 2 starts with a thorough review of the field.

Inference using a generative model involves the estimation of the conditional density $p(x|y)$. In this setting, any additional information on $p(x)$ is useful. As a simple example, assume that $p(x|y)$ is Gaussian. Then one can use the EM algorithm to find the parameters of the Gaussian corresponding to each class. The only difference to the standard EM algorithm as used for clustering is that the "hidden variable" associated with any labeled example is actually not hidden, but it is known and equals its class label. It implements the cluster assumption (cf. section 2.2.1), since a given cluster belongs to only one class.

mixture models

This small example already highlights different interpretations of semi-supervised learning with a generative model:

- It can be seen as classification with additional information on the marginal density.
- It can be seen as clustering with additional information. In the standard setting, this information would be the labels of a subset of points, but it could also come in the more general form of constraints. This is the topic of chapter 5.

A strength of the generative approach is that knowledge of the structure of the problem or the data can naturally be incorporated by modeling it. In chapter 3, this is demonstrated for the application of the EM algorithm to text data. It is observed that, when modeling assumptions are not correct, unlabeled data can decrease prediction accuracy. This effect is investigated in depth in chapter 4.

In statistical learning, before performing inference, one chooses a class of functions, or a prior over functions. One has to choose it according to what is known in advance about the problem. In the semi-supervised learning context, if one has some ideas about what the structure of the data tells about the target function, the

if you know your problem well, you can choose an appropriate statistical prior.

data-dependent priors

choice of this prior can be made more precise after seeing the unlabeled data: one could typically put a higher prior probability on functions that satisfy the cluster assumption. From a theoretical point, this is a natural way to obtain bounds for semi-supervised learning as explained in chapter 22.

II

1.3.2 Low-Density Separation

Part II of this book aims at describing algorithms which try to directly implement the low-density separation assumption by pushing the decision boundary away from the unlabeled points.

The most common approach to achieving this goal is to use a maximum margin algorithm such as support vector machines. The method of maximizing the margin for unlabeled as well as labeled points is called the transductive SVM (TSVM). However, the corresponding problem is nonconvex and thus difficult to optimize.

transductive SVM (TSVM)

One optimization algorithm for the TSVM is presented in chapter 6. Starting from the SVM solution as trained on the labeled data only, the unlabeled points are labeled by SVM predictions, and the SVM is retrained on all points. This is iterated while the weight of the unlabeled points is slowly increased. Another possibility is the semi-definite programming SDP relaxation suggested in chapter 7.

Two alternatives to the TSVM are then presented that are formulated in a probabilistic and in an information theoretic framework, respectively. In chapter 8, binary Gaussian process classification is augmented by the introduction of a null class that occupies the space between the two regular classes. As an advantage over the TSVM, this allows for probabilistic outputs.

This advantage is shared by the entropy minimization presented in chapter 9. It encourages the class-conditional probabilities $P(y|x)$ to be close to either 1 or 0 at labeled and unlabeled points. As a consequence of the smoothness assumption, the probability will tend to be close to 0 or 1 throughout any high-density region, while class boundaries correspond to intermediate probabilities.

A different way of using entropy or information is the data-dependent regularization developed in chapter 10. As compared to the TSVM, this seems to implement the low-density separation even more directly: the standard squared-norm regularizer is multiplied by a term reflecting the density close to the decision boundary.

III

1.3.3 Graph-Based Methods

represent samples as nodes, distances as weighted edges

During the last couple of years, the most active area of research in semi-supervised learning has been in graph-based methods, which are the topic of part III of this book. The common denominator of these methods is that the data are represented by the nodes of a graph, the edges of which are labeled with the pairwise distances of the incident nodes (and a missing edge corresponds to infinite distance). If the distance of two points is computed by minimizing the aggregate path distance over all paths connecting the two points, this can be seen as an approximation of the geodesic distance of the two points with respect to the manifold of data points.

if needed, watch Luxemburg lecture

Thus, graph methods can be argued to build on the manifold assumption.

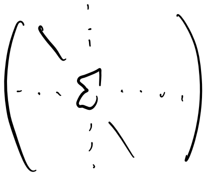
weight matrix

Most graph methods refer to the graph by utilizing the graph Laplacian. Let $\mathsf{g} = (V, E)$ be a graph with real edge weights given by $w : E \to \mathbb{R}$. Here, the weight $w(e)$ of an edge e indicates the similarity of the incident nodes (and a missing edge corresponds to zero similarity). Now the weighted adjacency matrix (or weight matrix, for short) $\mathbf{W}$ of the graph $\mathsf{g} = (V, E)$ is defined by

$$\mathbf{W}_{ij} := \begin{cases} w(e) & \text{if } e = (i,j) \in E, \\ 0 & \text{if } e = (i,j) \in E. \end{cases} \tag{1.2}$$

graph Laplacian

The diagonal matrix $\mathbf{D}$ defined by $\mathbf{D}_{ii} := \sum_j W_{ij}$ is called the degree matrix of g. Now there are different ways of defining the graph Laplacian, the two most prominent of which are the normalized graph Laplacian, $\mathcal{L}$, and the unnormalized graph Laplacian, L:



$$\begin{aligned} \mathcal{L} &:= \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \\ L &:= \mathbf{D} - \mathbf{W}. \end{aligned} \tag{1.3}$$

Many graph methods that penalize nonsmoothness along the edges of a weighted graph can in retrospect be seen as different instances of a rather general family of algorithms, as is outlined in chapter 11. Chapter 13 takes a more theoretical point of view, and transfers notions of smoothness from the continuous case onto graphs as the discrete case. From that, it proposes different regularizers based on a graph representation of the data.

Usually the prediction consists of labels for the unlabeled nodes. For this reason, this kind of algorithm is intrinsically transductive, i.e., it returns only the value of the decision function on the unlabeled points and not the decision function itself. However, there has been recent work in order to extend graph-based methods to produce inductive solutions, as discussed in chapter 12.

Information propagation on graphs can also serve to improve a given (possibly strictly supervised) classification, taking unlabeled data into account. Chapter 14 presents a probabilistic method for using directed graphs in this manner.

Often the graph g is constructed by computing similarities of objects in some other representation, e.g., using a kernel function on Euclidean data points. But sometimes the original data already have the form of a graph. Examples include the linkage pattern of webpages and the interactions of proteins (see chapter 20). In such cases, the directionality of the edges may be important.

IV

1.3.4 Change of Representation

first unsupervised, then supervised

The topic of part IV is algorithms that are not intrinsically semi-supervised, but instead perform two-step learning:

1. Perform an unsupervised step on all data, labeled and unlabeled, but ignoring the available labels. This can, for instance, be a change of representation, or the

construction of a new metric or a new kernel.

2. Ignore the unlabeled data and perform plain supervised learning using the new distance, representation, or kernel.

This can be seen as direct implementation of the semi-supervised smoothness assumption, since the representation is changed in such a way that small distances in high-density regions are conserved.

Note that the graph-based methods (part III) are closely related to the ones presented in this part: the very construction of the graph from the data can be seen as an unsupervised change of representation. Consequently, the first chapter of part IV, chapter 15, discusses spectral transforms of such graphs in order to build kernels. Spectral methods can also be used for nonlinear dimensionality reduction, as extended in chapter 16. Furthermore, in chapter 17, metrics derived from graphs are investigated, for example, those derived from shortest paths.

1.3.5 Semi-Supervised Learning in Practice

Semi-supervised learning will be most useful whenever there are far more unlabeled data than labeled. This is likely to occur if obtaining data points is cheap, but obtaining the labels costs a lot of time, effort, or money. This is the case in many application areas of machine learning, for example:

- In speech recognition, it costs almost nothing to record huge amounts of speech, but labeling it requires some human to listen to it and type a transcript.
- Billions of webpages are directly available for automated processing, but to classify them reliably, humans have to read them.
- Protein sequences are nowadays acquired at industrial speed (by genome sequencing, computational gene finding, and automatic translation), but to resolve a three-dimensional (3D) structure or to determine the functions of a single protein may require years of scientific work.

text and speech processing. More generally, everything on which we have lots of data (video, foto, maps, ...)

Webpage classification is introduced in chapter 3 in the context of generative models.

Since unlabeled data carry less information than labeled data, they are required in large amounts in order to increase prediction accuracy significantly. This implies the need for fast and efficient SSL algorithms. Chapters 18 and 19 present two approaches to dealing with huge numbers of points. In chapter 18 methods are developed for speeding up the label propagation methods introduced in chapter 11. In chapter 19 cluster kernels are shown to be an efficient SSL method.

Chapter 19 also presents the first of two approaches to an important bioinformatics application of semi-supervised learning: the classification of protein sequences. While here the predictions are based on the protein sequences themselves, Chapter 20 moves on to a somewhat more complex setting: The information is here assumed to be present in the form of graphs that characterize the interactions of proteins. Several such graphs exist and have to be combined in an appropriate way.

tips for practioners

This book part concludes with a very practical chapter: the presentation and evaluation of the benchmarks associated with this book (chapter 21). It is intended to give hints to the practitioner on how to choose suitable methods based on the properties of the problem.

1.3.6 Outlook

The last part of the book, part VI, is devoted to some of the most interesting directions of ongoing research in SSL.

Until now this book has mostly resticted itself to classification. Chapter 23 introduces another approach to SSL that is suited for both classification and regression, and derives algorithms from it. Interestingly it seems not to require the assumptions proposed in chapter 1.

Further, this book mostly presented *algorithms* for SSL. While the assumptions discussed above supply some intuition on when and why SSL works, and chapter 4 investigates when and why it can fail, it would clearly be more satisfactory to have a thorough theoretical understanding of SSL in total. Chapter 22 offers a PAC-style framework that yields error bounds for SSL problems.

In chapter 24 inductive semi-supervised learning and transduction are compared in terms of Vapnik-Chervonenkis (VC) bounds and other theoretical and philosophical concepts.

The book closes with a hypothetical discussion (chapter 25) between three machine learning researchers on the relationship of (and the differences between) semi-supervised learning and transduction.