24 Generative Models

We started this book with a *distribution free* learning framework; namely, we did not impose any assumptions on the underlying distribution over the data. Furthermore, we followed a *discriminative* approach in which our goal is not to learn the underlying distribution but rather to learn an accurate predictor. In this chapter we describe a *generative* approach, in which it is assumed that the underlying distribution over the data has a specific parametric form and our goal is to estimate the parameters of the model. This task is called *parametric density estimation*.

The discriminative approach has the advantage of directly optimizing the quantity of interest (the prediction accuracy) instead of learning the underlying distribution. This was phrased as follows by Vladimir Vapnik in his principle for solving problems using a restricted amount of information:

When solving a given problem, try to avoid a more general problem as an intermediate step.

Of course, if we succeed in learning the underlying distribution accurately, we are considered to be "experts" in the sense that we can predict by using the Bayes optimal classifier. The problem is that it is usually more difficult to learn the underlying distribution than to learn an accurate predictor. However, in some situations, it is reasonable to adopt the generative learning approach. For example, sometimes it is easier (computationally) to estimate the parameters of the model than to learn a discriminative predictor. Additionally, in some cases we do not have a specific task at hand but rather would like to model the data either for making predictions at a later time without having to retrain a predictor or for the sake of interpretability of the data.

We start with a popular statistical method for estimating the parameters of the data, which is called the maximum likelihood principle. Next, we describe two generative assumptions which greatly simplify the learning process. We also describe the EM algorithm for calculating the maximum likelihood in the presence of latent variables. We conclude with a brief description of Bayesian reasoning.

24.1 Maximum Likelihood Estimator

Let us start with a simple example. A drug company developed a new drug to treat some deadly disease. We would like to estimate the probability of survival when using the drug. To do so, the drug company sampled a training set of m people and gave them the drug. Let $S = (x_1, \ldots, x_m)$ denote the training set, where for each i, $x_i = 1$ if the ith person survived and $x_i = 0$ otherwise. We can model the underlying distribution using a single parameter, $\theta \in [0, 1]$, indicating the probability of survival.

We now would like to estimate the parameter θ on the basis of the training set S. A natural idea is to use the average number of 1's in S as an estimator. That is,

$$\hat{\theta} = \frac{1}{m} \sum_{i=1}^{m} x_i. \tag{24.1}$$

Clearly, $\mathbb{E}_S[\hat{\theta}] = \theta$. That is, $\hat{\theta}$ is an *unbiased estimator* of θ. Furthermore, since $\hat{\theta}$ is the average of m i.i.d. binary random variables we can use Hoeffding's inequality to get that with probability of at least $1 - \delta$ over the choice of S we have that

$$|\hat{\theta} - \theta| \;\leq\; \sqrt{\frac{\log(2/\delta)}{2\,m}}. \tag{24.2}$$

Another interpretation of $\hat{\theta}$ is as the *Maximum Likelihood Estimator*, as we formally explain now. We first write the probability of generating the sample S:

$$\mathbb{P}[S = (x_1, \ldots, x_m)] = \prod_{i=1}^{m} \theta^{x_i} \, (1-\theta)^{1-x_i} = \theta^{\sum_i x_i} \, (1-\theta)^{\sum_i (1-x_i)}.$$

We define the *log likelihood* of S, given the parameter θ, as the log of the preceding expression:

$$L(S;\theta) = \log\left(\mathbb{P}[S = (x_1, \ldots, x_m)]\right) = \log(\theta) \sum_i x_i + \log(1-\theta) \sum_i (1 - x_i).$$

The maximum likelihood estimator is the parameter that maximizes the likelihood

$$\hat{\theta} \in \operatorname*{argmax}_{\theta} L(S;\theta). \tag{24.3}$$

Next, we show that in our case, Equation (24.1) is a maximum likelihood estimator. To see this, we take the derivative of $L(S;\theta)$ with respect to θ and equate it to zero:

$$\frac{\sum_i x_i}{\theta} - \frac{\sum_i (1 - x_i)}{1 - \theta} = 0.$$

Solving the equation for θ we obtain the estimator given in Equation (24.1).

24.1.1 Maximum Likelihood Estimation for Continuous Random Variables

Let X be a continuous random variable. Then, for most $x \in \mathbb{R}$ we have $\mathbb{P}[X = x] = 0$ and therefore the definition of likelihood as given before is trivialized. To overcome this technical problem we define the likelihood as log of the *density* of the probability of X at x. That is, given an i.i.d. training set $S = (x_1, \ldots, x_m)$ sampled according to a density distribution $\mathcal{P}_\theta$ we define the likelihood of S given θ as

$$L(S;\theta) = \log\left(\prod_{i=1}^{m} \mathcal{P}_\theta(x_i)\right) = \sum_{i=1}^{m} \log(\mathcal{P}_\theta(x_i)).$$

As before, the maximum likelihood estimator is a maximizer of $L(S;\theta)$ with respect to θ.

As an example, consider a Gaussian random variable, for which the density function of X is parameterized by $\theta = (\mu, \sigma)$ and is defined as follows:

$$\mathcal{P}_\theta(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

We can rewrite the likelihood as

$$L(S;\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i - \mu)^2 - m\log(\sigma\sqrt{2\pi}).$$

To find a parameter $\theta = (\mu, \sigma)$ that optimizes this we take the derivative of the likelihood w.r.t. μ and w.r.t. σ and compare it to 0. We obtain the following two equations:

$$\frac{d}{d\mu}L(S;\theta) = \frac{1}{\sigma^2}\sum_{i=1}^{m}(x_i - \mu) = 0$$
$$\frac{d}{d\sigma}L(S;\theta) = \frac{1}{\sigma^3}\sum_{i=1}^{m}(x_i - \mu)^2 - \frac{m}{\sigma} = 0$$

Solving the preceding equations we obtain the maximum likelihood estimates:

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{and} \qquad \hat{\sigma} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu})^2}$$

Note that the maximum likelihood estimate is not always an unbiased estimator. For example, while $\hat{\mu}$ is unbiased, it is possible to show that the estimate $\hat{\sigma}$ of the variance is biased (Exercise 1).

Simplifying Notation

To simplify our notation, we use $\mathcal{P}[X = x]$ in this chapter to describe both the probability that $X = x$ (for discrete random variables) and the density of the distribution at x (for continuous variables).

24.1.2 Maximum Likelihood and Empirical Risk Minimization

The maximum likelihood estimator shares some similarity with the Empirical Risk Minimization (ERM) principle, which we studied extensively in previous chapters. Recall that in the ERM principle we have a hypothesis class $\mathcal{H}$ and we use the training set for choosing a hypothesis $h \in \mathcal{H}$ that minimizes the empirical risk. We now show that the maximum likelihood estimator is an ERM for a particular loss function.

Given a parameter θ and an observation x, we define the loss of θ on x as

$$\ell(\theta, x) = -\log(\mathcal{P}_\theta[x]). \tag{24.4}$$

That is, $\ell(\theta, x)$ is the negation of the log-likelihood of the observation x, assuming the data is distributed according to $\mathcal{P}_\theta$. This loss function is often referred to as the log-loss. On the basis of this definition it is immediate that the maximum likelihood principle is equivalent to minimizing the empirical risk with respect to the loss function given in Equation (24.4). That is,

$$\operatorname*{argmin}_\theta \sum_{i=1}^{m} (-\log(\mathcal{P}_\theta[x_i])) = \operatorname*{argmax}_\theta \sum_{i=1}^{m} \log(\mathcal{P}_\theta[x_i]).$$

Assuming that the data is distributed according to a distribution $\mathcal{P}$ (not necessarily of the parametric form we employ), the true risk of a parameter θ becomes

$$\begin{aligned} \mathop{\mathbb{E}}_x[\ell(\theta, x)] &= -\sum_x \mathcal{P}[x] \log(\mathcal{P}_\theta[x]) \\ &= \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{\mathcal{P}[x]}{\mathcal{P}_\theta[x]}\right)}_{D_{\mathrm{RE}}[\mathcal{P}||\mathcal{P}_\theta]} + \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{1}{\mathcal{P}[x]}\right)}_{H(\mathcal{P})}, \end{aligned} \tag{24.5}$$

where D_{RE} is called the *relative entropy*, and H is called the *entropy function*. The relative entropy is a divergence measure between two probabilities. For discrete variables, it is always nonnegative and is equal to 0 only if the two distributions are the same. It follows that the true risk is minimal when $\mathcal{P}_\theta = \mathcal{P}$.

The expression given in Equation (24.5) underscores how our generative assumption affects our density estimation, even in the limit of infinite data. It shows that if the underlying distribution is indeed of a parametric form, then by choosing the correct parameter we can make the risk be the entropy of the distribution. However, if the distribution is not of the assumed parametric form, even the best parameter leads to an inferior model and the suboptimality is measured by the relative entropy divergence.

24.1.3 Generalization Analysis

How good is the maximum likelihood estimator when we learn from a finite training set?

To answer this question we need to define how we assess the quality of an approximated solution of the density estimation problem. Unlike discriminative learning, where there is a clear notion of "loss," in generative learning there are various ways to define the loss of a model. On the basis of the previous subsection, one natural candidate is the expected log-loss as given in Equation (24.5).

In some situations, it is easy to prove that the maximum likelihood principle guarantees low true risk as well. For example, consider the problem of estimating the mean of a Gaussian variable of unit variance. We saw previously that the maximum likelihood estimator is the average: $\hat{\mu} = \frac{1}{m}\sum_i x_i$. Let $\mu^\star$ be the optimal parameter. Then,

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)}[\ell(\hat{\mu},x) - \ell(\mu^\star,x)] &= \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)} \log\left(\frac{\mathcal{P}_{\mu^\star}[x]}{\mathcal{P}_{\hat{\mu}}[x]}\right) \\
&= \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)} \left(-\frac{1}{2}(x-\mu^\star)^2 + \frac{1}{2}(x-\hat{\mu})^2\right) \\
&= \frac{\hat{\mu}^2}{2} - \frac{(\mu^\star)^2}{2} + (\mu^\star - \hat{\mu}) \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)}[x] \\
&= \frac{\hat{\mu}^2}{2} - \frac{(\mu^\star)^2}{2} + (\mu^\star - \hat{\mu})\,\mu^\star \\
&= \frac{1}{2}(\hat{\mu} - \mu^\star)^2. \qquad (24.6)
\end{aligned}$$

Next, we note that $\hat{\mu}$ is the average of m Gaussian variables and therefore it is also distributed normally with mean $\mu^\star$ and variance $\sigma^\star/m$. From this fact we can derive bounds of the form: with probability of at least $1-\delta$ we have that $|\hat{\mu} - \mu^\star| \leq \epsilon$ where ϵ depends on $\sigma^\star/m$ and on δ.

In some situations, the maximum likelihood estimator clearly overfits. For example, consider a Bernoulli random variable X and let $\mathcal{P}[X=1] = \theta^\star$. As we saw previously, using Hoeffding's inequality we can easily derive a guarantee on $|\theta^\star - \hat{\theta}|$ that holds with high probability (see Equation (24.2)). However, if our goal is to obtain a small value of the expected log-loss function as defined in Equation (24.5) we might fail. For example, assume that $\theta^\star$ is nonzero but very small. Then, the probability that no element of a sample of size m will be 1 is $(1-\theta^\star)^m$, which is greater than $e^{-2\theta^\star m}$. It follows that whenever $m \leq \frac{\log(2)}{2\theta^\star}$, the probability that the sample is all zeros is at least 50%, and in that case, the maximum likelihood rule will set $\hat{\theta} = 0$. But the true risk of the estimate $\hat{\theta} = 0$ is

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\theta^\star}[\ell(\hat{\theta},x)] &= \theta^\star \ell(\hat{\theta},1) + (1-\theta^\star)\ell(\hat{\theta},0) \\
&= \theta^\star \log(1/\hat{\theta}) + (1-\theta^\star)\log(1/(1-\hat{\theta})) \\
&= \theta^\star \log(1/0) = \infty.
\end{aligned}$$

This simple example shows that we should be careful in applying the maximum likelihood principle.

To overcome overfitting, we can use the variety of tools we encountered pre-

viously in the book. A simple regularization technique is outlined in Exercise 2.

24.2 Naive Bayes

The Naive Bayes classifier is a classical demonstration of how generative assumptions and parameter estimations simplify the learning process. Consider the problem of predicting a label $y \in \{0,1\}$ on the basis of a vector of features $\mathbf{x} = (x_1, \ldots, x_d)$, where we assume that each x_i is in $\{0,1\}$. Recall that the Bayes optimal classifier is

$$h_{\text{Bayes}}(\mathbf{x}) = \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y | X = \mathbf{x}].$$

To describe the probability function $\mathcal{P}[Y = y|X = \mathbf{x}]$ we need 2^d parameters, each of which corresponds to $\mathcal{P}[Y = 1|X = \mathbf{x}]$ for a certain value of $\mathbf{x} \in \{0,1\}^d$. This implies that the number of examples we need grows exponentially with the number of features.

In the Naive Bayes approach we make the (rather naive) generative assumption that given the label, the features are independent of each other. That is,

$$\mathcal{P}[X = \mathbf{x}|Y = y] = \prod_{i=1}^{d} \mathcal{P}[X_i = x_i|Y = y].$$

With this assumption and using Bayes' rule, the Bayes optimal classifier can be further simplified:

$$\begin{aligned} h_{\text{Bayes}}(\mathbf{x}) &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y|X = \mathbf{x}] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y]\mathcal{P}[X = \mathbf{x}|Y = y] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y] \prod_{i=1}^{d} \mathcal{P}[X_i = x_i|Y = y]. \end{aligned} \tag{24.7}$$

That is, now the number of parameters we need to estimate is only $2d + 1$. Here, the generative assumption we made reduced significantly the number of parameters we need to learn.

When we also estimate the parameters using the maximum likelihood principle, the resulting classifier is called the *Naive Bayes* classifier.

24.3 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is another demonstration of how generative assumptions simplify the learning process. As in the Naive Bayes classifier we consider again the problem of predicting a label $y \in \{0,1\}$ on the basis of a

vector of features $\mathbf{x} = (x_1, \ldots, x_d)$. But now the generative assumption is as follows. First, we assume that $\mathcal{P}[Y = 1] = \mathcal{P}[Y = 0] = 1/2$. Second, we assume that the conditional probability of X given Y is a Gaussian distribution. Finally, the covariance matrix of the Gaussian distribution is the same for both values of the label. Formally, let $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1 \in \mathbb{R}^d$ and let Σ be a covariance matrix. Then, the density distribution is given by

$$\mathcal{P}[X = \mathbf{x}|Y = y] = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_y)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_y)\right).$$

As we have shown in the previous section, using Bayes' rule we can write

$$h_{\text{Bayes}}(\mathbf{x}) = \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y]\mathcal{P}[X = \mathbf{x}|Y = y].$$

This means that we will predict $h_{\text{Bayes}}(\mathbf{x}) = 1$ iff

$$\log\left(\frac{\mathcal{P}[Y = 1]\mathcal{P}[X = \mathbf{x}|Y = 1]}{\mathcal{P}[Y = 0]\mathcal{P}[X = \mathbf{x}|Y = 0]}\right) > 0.$$

This ratio is often called the *log-likelihood ratio*.

In our case, the log-likelihood ratio becomes

$$\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_0) - \tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_1)$$

We can rewrite this as $\langle \mathbf{w}, \mathbf{x} \rangle + b$ where

$$\mathbf{w} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma^{-1} \quad \text{and} \quad b = \tfrac{1}{2}\left(\boldsymbol{\mu}_0^T \Sigma^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T \Sigma^{-1} \boldsymbol{\mu}_1\right). \tag{24.8}$$

As a result of the preceding derivation we obtain that under the aforementioned generative assumptions, the Bayes optimal classifier is a linear classifier. Additionally, one may train the classifier by estimating the parameter $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ and Σ from the data, using, for example, the maximum likelihood estimator. With those estimators at hand, the values of $\mathbf{w}$ and b can be calculated as in Equation (24.8).

24.4 Latent Variables and the EM Algorithm

In generative models we assume that the data is generated by sampling from a specific parametric distribution over our instance space $\mathcal{X}$. Sometimes, it is convenient to express this distribution using latent random variables. A natural example is a mixture of k Gaussian distributions. That is, $\mathcal{X} = \mathbb{R}^d$ and we assume that each $\mathbf{x}$ is generated as follows. First, we choose a random number in $\{1, \ldots, k\}$. Let Y be a random variable corresponding to this choice, and denote $\mathcal{P}[Y = y] = c_y$. Second, we choose $\mathbf{x}$ on the basis of the value of Y according to a Gaussian distribution

$$\mathcal{P}[X = \mathbf{x}|Y = y] = \frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_y)^T \Sigma_y^{-1}(\mathbf{x} - \boldsymbol{\mu}_y)\right). \tag{24.9}$$

Therefore, the density of X can be written as:

$$\begin{aligned}\mathcal{P}[X = \mathbf{x}] &= \sum_{y=1}^{k} \mathcal{P}[Y = y]\mathcal{P}[X = \mathbf{x}|Y = y] \\ &= \sum_{y=1}^{k} c_y \frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_y)^T \Sigma_y^{-1}(\mathbf{x} - \boldsymbol{\mu}_y)\right).\end{aligned}$$

Note that Y is a hidden variable that we do not observe in our data. Nevertheless, we introduce Y since it helps us describe a simple parametric form of the probability of X.

More generally, let $\boldsymbol{\theta}$ be the parameters of the joint distribution of X and Y (e.g., in the preceding example, $\boldsymbol{\theta}$ consists of c_y, $\boldsymbol{\mu}_y$, and Σ_y, for all $y = 1, \ldots, k$). Then, the log-likelihood of an observation $\mathbf{x}$ can be written as

$$\log\left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}]\right) = \log\left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}, Y = y]\right).$$

Given an i.i.d. sample, $S = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$, we would like to find $\boldsymbol{\theta}$ that maximizes the log-likelihood of S,

$$\begin{aligned}L(\boldsymbol{\theta}) &= \log \prod_{i=1}^{m} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \\ &= \sum_{i=1}^{m} \log \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \\ &= \sum_{i=1}^{m} \log\left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]\right).\end{aligned}$$

The maximum-likelihood estimator is therefore the solution of the maximization problem

$$\operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log\left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]\right).$$

In many situations, the summation inside the log makes the preceding optimization problem computationally hard. The *Expectation-Maximization* (EM) algorithm, due to Dempster, Laird, and Rubin, is an iterative procedure for searching a (local) maximum of $L(\boldsymbol{\theta})$. While EM is not guaranteed to find the global maximum, it often works reasonably well in practice.

EM is designed for those cases in which, had we known the values of the latent variables Y, then the maximum likelihood optimization problem would have been tractable. More precisely, define the following function over $m \times k$ matrices and the set of parameters $\boldsymbol{\theta}$:

$$F(Q, \boldsymbol{\theta}) = \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log\left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]\right).$$

If each row of Q defines a probability over the ith latent variable given $X = \mathbf{x}_i$, then we can interpret $F(Q, \boldsymbol{\theta})$ as the expected log-likelihood of a training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where the expectation is with respect to the choice of each y_i on the basis of the ith row of Q. In the definition of F, the summation is outside the log, and we assume that this makes the optimization problem with respect to $\boldsymbol{\theta}$ tractable:

ASSUMPTION 24.1 For any matrix $Q \in [0,1]^{m,k}$, such that each row of Q sums to 1, the optimization problem

$$\operatorname*{argmax}_{\boldsymbol{\theta}} F(Q, \boldsymbol{\theta})$$

is tractable.

The intuitive idea of EM is that we have a "chicken and egg" problem. On one hand, had we known Q, then by our assumption, the optimization problem of finding the best $\boldsymbol{\theta}$ is tractable. On the other hand, had we known the parameters $\boldsymbol{\theta}$ we could have set $Q_{i,y}$ to be the probability of $Y = y$ given that $X = \mathbf{x}_i$. The EM algorithm therefore alternates between finding $\boldsymbol{\theta}$ given Q and finding Q given $\boldsymbol{\theta}$. Formally, EM finds a sequence of solutions $(Q^{(1)}, \boldsymbol{\theta}^{(1)}), (Q^{(2)}, \boldsymbol{\theta}^{(2)}), \ldots$ where at iteration t, we construct $(Q^{(t+1)}, \boldsymbol{\theta}^{(t+1)})$ by performing two steps.

- **E**xpectation Step: Set

$$Q_{i,y}^{(t+1)} = \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y | X = \mathbf{x}_i]. \tag{24.10}$$

  This step is called the Expectation step, because it yields a new probability over the latent variables, which defines a new *expected* log-likelihood function over $\boldsymbol{\theta}$.
- **M**aximization Step: Set $\boldsymbol{\theta}^{(t+1)}$ to be the maximizer of the expected log-likelihood, where the expectation is according to $Q^{(t+1)}$:

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)}, \boldsymbol{\theta}). \tag{24.11}$$

  By our assumption, it is possible to solve this optimization problem efficiently.

The initial values of $\boldsymbol{\theta}^{(1)}$ and $Q^{(1)}$ are usually chosen at random and the procedure terminates after the improvement in the likelihood value stops being significant.

24.4.1 EM as an Alternate Maximization Algorithm

To analyze the EM algorithm, we first view it as an alternate maximization algorithm. Define the following objective function

$$G(Q, \boldsymbol{\theta}) = F(Q, \boldsymbol{\theta}) - \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log(Q_{i,y}).$$

The second term is the sum of the *entropies* of the rows of Q. Let

$$\mathbb{Q} = \left\{ Q \in [0,1]^{m,k} : \forall i,\ \sum_{y=1}^{k} Q_{i,y} = 1 \right\}$$

be the set of matrices whose rows define probabilities over $[k]$. The following lemma shows that EM performs alternate maximization iterations for maximizing G.

LEMMA 24.2 *The EM procedure can be rewritten as:*

$$\begin{aligned} Q^{(t+1)} &= \operatorname*{argmax}_{Q \in \mathbb{Q}} G(Q, \boldsymbol{\theta}^{(t)}) \\ \boldsymbol{\theta}^{(t+1)} &= \operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)}, \boldsymbol{\theta}) \ . \end{aligned}$$

Furthermore, $G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$.

Proof Given $Q^{(t+1)}$ we clearly have that

$$\operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)}, \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)}, \boldsymbol{\theta}).$$

Therefore, we only need to show that for any $\boldsymbol{\theta}$, the solution of $\operatorname{argmax}_{Q \in \mathbb{Q}} G(Q, \boldsymbol{\theta})$ is to set $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y | X = \mathbf{x}_i]$. Indeed, by Jensen's inequality, for any $Q \in \mathbb{Q}$ we have that

$$\begin{aligned} G(Q, \boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(\sum_{y=1}^{k} Q_{i,y} \log \left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{Q_{i,y}} \right) \right) \\ &\leq \sum_{i=1}^{m} \left(\log \left(\sum_{y=1}^{k} Q_{i,y} \frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{Q_{i,y}} \right) \right) \\ &= \sum_{i=1}^{m} \log \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y] \right) \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) = L(\boldsymbol{\theta}), \end{aligned}$$

while for $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]$ we have

$$\begin{aligned} G(Q, \boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \log \left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{\mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]} \right) \right) \\ &= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) \sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) = L(\boldsymbol{\theta}). \end{aligned}$$

This shows that setting $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]$ maximizes $G(Q, \boldsymbol{\theta})$ over $Q \in \mathbb{Q}$ and shows that $G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$. □

The preceding lemma immediately implies:

THEOREM 24.3 *The EM procedure never decreases the log-likelihood; namely, for all t,*

$$L(\boldsymbol{\theta}^{(t+1)}) \geq L(\boldsymbol{\theta}^{(t)}).$$

Proof By the lemma we have

$$L(\boldsymbol{\theta}^{(t+1)}) = G(Q^{(t+2)}, \boldsymbol{\theta}^{(t+1)}) \geq G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)}).$$

□

24.4.2 EM for Mixture of Gaussians (Soft k-Means)

Consider the case of a mixture of k Gaussians in which $\boldsymbol{\theta}$ is a triplet $(\mathbf{c}, \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}, \{\Sigma_1, \ldots, \Sigma_k\})$ where $\mathcal{P}_{\boldsymbol{\theta}}[Y = y] = c_y$ and $\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}|Y = y]$ is as given in Equation (24.9). For simplicity, we assume that $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = I$, where I is the identity matrix. Specifying the EM algorithm for this case we obtain the following:

- **E**xpectation step: For each $i \in [m]$ and $y \in [k]$ we have that

$$\begin{aligned} \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y|X = \mathbf{x}_i] &= \frac{1}{Z_i} \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y] \, \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[X = \mathbf{x}_i|Y = y] \\ &= \frac{1}{Z_i} c_y^{(t)} \exp \left(-\frac{1}{2} \|\mathbf{x}_i - \boldsymbol{\mu}_y^{(t)}\|^2 \right), \end{aligned} \tag{24.12}$$

  where Z_i is a normalization factor which ensures that $\sum_y \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y|X = \mathbf{x}_i]$ sums to 1.
- **M**aximization step: We need to set $\boldsymbol{\theta}^{t+1}$ to be a maximizer of Equation (24.11),

which in our case amounts to maximizing the following expression w.r.t. $\mathbf{c}$ and $\boldsymbol{\mu}$:

$$\sum_{i=1}^{m}\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]\left(\log(c_y)-\frac{1}{2}\|\mathbf{x}_i-\boldsymbol{\mu}_y\|^2\right). \tag{24.13}$$

Comparing the derivative of Equation (24.13) w.r.t. $\boldsymbol{\mu}_y$ to zero and rearranging terms we obtain:

$$\boldsymbol{\mu}_y = \frac{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]\,\mathbf{x}_i}{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]}.$$

That is, $\boldsymbol{\mu}_y$ is a weighted average of the $\mathbf{x}_i$ where the weights are according to the probabilities calculated in the E step. To find the optimal $\mathbf{c}$ we need to be more careful since we must ensure that $\mathbf{c}$ is a probability vector. In Exercise 3 we show that the solution is:

$$c_y = \frac{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]}{\sum_{y'=1}^{k}\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y'|X=\mathbf{x}_i]}. \tag{24.14}$$

It is interesting to compare the preceding algorithm to the k-means algorithm described in Chapter 22. In the k-means algorithm, we first assign each example to a cluster according to the distance $\|\mathbf{x}_i-\boldsymbol{\mu}_y\|$. Then, we update each center $\boldsymbol{\mu}_y$ according to the average of the examples assigned to this cluster. In the EM approach, however, we determine the probability that each example belongs to each cluster. Then, we update the centers on the basis of a weighted sum over the entire sample. For this reason, the EM approach for k-means is sometimes called "soft k-means."

24.5 Bayesian Reasoning

The maximum likelihood estimator follows a frequentist approach. This means that we refer to the parameter θ as a fixed parameter and the only problem is that we do not know its value. A different approach to parameter estimation is called Bayesian reasoning. In the Bayesian approach, our uncertainty about θ is also modeled using probability theory. That is, we think of θ as a random variable as well and refer to the distribution $\mathcal{P}[\theta]$ as a *prior distribution*. As its name indicates, the prior distribution should be defined by the learner prior to observing the data.

As an example, let us consider again the drug company which developed a new drug. On the basis of past experience, the statisticians at the drug company believe that whenever a drug has reached the level of clinic experiments on people, it is likely to be effective. They model this prior belief by defining a density distribution on θ such that

$$\mathcal{P}[\theta]=\begin{cases}0.8 & \text{if } \theta>0.5\\ 0.2 & \text{if } \theta\leq 0.5\end{cases} \tag{24.15}$$

As before, given a specific value of θ, it is assumed that the conditional probability, $\mathcal{P}[X = x|\theta]$, is known. In the drug company example, X takes values in $\{0, 1\}$ and $\mathcal{P}[X = x|\theta] = \theta^x(1-\theta)^{1-x}$.

Once the prior distribution over θ and the conditional distribution over X given θ are defined, we again have complete knowledge of the distribution over X. This is because we can write the probability over X as a marginal probability

$$\mathcal{P}[X = x] = \sum_{\theta} \mathcal{P}[X = x, \theta] = \sum_{\theta} \mathcal{P}[\theta]\mathcal{P}[X = x|\theta],$$

where the last equality follows from the definition of conditional probability. If θ is continuous we replace $\mathcal{P}[\theta]$ with the density function and the sum becomes an integral:

$$\mathcal{P}[X = x] = \int_{\theta} \mathcal{P}[\theta]\mathcal{P}[X = x|\theta]\, d\theta.$$

Seemingly, once we know $\mathcal{P}[X = x]$, a training set $S = (x_1, \ldots, x_m)$ tells us nothing as we are already experts who know the distribution over a new point X. However, the Bayesian view introduces dependency between S and X. This is because we now refer to θ as a random variable. A new point X and the previous points in S are independent *only* conditioned on θ. This is different from the frequentist philosophy in which θ is a parameter that we might not know, but since it is just a parameter of the distribution, a new point X and previous points S are always independent.

In the Bayesian framework, since X and S are not independent anymore, what we would like to calculate is the probability of X given S, which by the chain rule can be written as follows:

$$\mathcal{P}[X = x|S] = \sum_{\theta} \mathcal{P}[X = x|\theta, S]\, \mathcal{P}[\theta|S] = \sum_{\theta} \mathcal{P}[X = x|\theta]\, \mathcal{P}[\theta|S].$$

The second inequality follows from the assumption that X and S are independent when we condition on θ. Using *Bayes' rule* we have

$$\mathcal{P}[\theta|S] = \frac{\mathcal{P}[S|\theta]\, \mathcal{P}[\theta]}{\mathcal{P}[S]},$$

and together with the assumption that points are independent conditioned on θ, we can write

$$\mathcal{P}[\theta|S] = \frac{\mathcal{P}[S|\theta]\, \mathcal{P}[\theta]}{\mathcal{P}[S]} = \frac{1}{\mathcal{P}[S]} \prod_{i=1}^{m} \mathcal{P}[X = x_i|\theta]\, \mathcal{P}[\theta].$$

We therefore obtain the following expression for Bayesian prediction:

$$\mathcal{P}[X = x|S] = \frac{1}{\mathcal{P}[S]} \sum_{\theta} \mathcal{P}[X = x|\theta] \prod_{i=1}^{m} \mathcal{P}[X = x_i|\theta]\, \mathcal{P}[\theta]. \tag{24.16}$$

Getting back to our drug company example, we can rewrite $\mathcal{P}[X = x|S]$ as

$$\mathcal{P}[X = x|S] = \frac{1}{P[S]} \int \theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i (1-x_i)}\, \mathcal{P}[\theta]\, d\theta.$$

It is interesting to note that when $\mathcal{P}[\theta]$ is uniform we obtain that

$$\mathcal{P}[X = x|S] \propto \int \theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i(1-x_i)}\, d\theta.$$

Solving the preceding integral (using integration by parts) we obtain

$$\mathcal{P}[X = 1|S] = \frac{(\sum_i x_i) + 1}{m + 2}.$$

Recall that the prediction according to the maximum likelihood principle in this case is $\mathcal{P}[X = 1|\hat{\theta}] = \frac{\sum_i x_i}{m}$. The Bayesian prediction with uniform prior is rather similar to the maximum likelihood prediction, except it adds "pseudoexamples" to the training set, thus biasing the prediction toward the uniform prior.

Maximum A Posteriori

In many situations, it is difficult to find a closed form solution to the integral given in Equation (24.16). Several numerical methods can be used to approximate this integral. Another popular solution is to find a single θ which maximizes $\mathcal{P}[\theta|S]$. The value of θ which maximizes $\mathcal{P}[\theta|S]$ is called the *Maximum A Posteriori* estimator. Once this value is found, we can calculate the probability that $X = x$ given the maximum *a posteriori* estimator and independently on S.

24.6 Summary

In the generative approach to machine learning we aim at modeling the distribution over the data. In particular, in parametric density estimation we further assume that the underlying distribution over the data has a specific parametric form and our goal is to estimate the parameters of the model. We have described several principles for parameter estimation, including maximum likelihood, Bayesian estimation, and maximum *a posteriori*. We have also described several specific algorithms for implementing the maximum likelihood under different assumptions on the underlying data distribution, in particular, Naive Bayes, LDA, and EM.

24.7 Bibliographic Remarks

The maximum likelihood principle was studied by Ronald Fisher in the beginning of the 20th century. Bayesian statistics follow Bayes' rule, which is named after the 18th century English mathematician Thomas Bayes.

There are many excellent books on the generative and Bayesian approaches to machine learning. See, for example, (Bishop 2006, Koller & Friedman 2009, MacKay 2003, Murphy 2012, Barber 2012).

24.8 Exercises

1. Prove that the maximum likelihood estimator of the variance of a Gaussian variable is biased.
2. Regularization for Maximum Likelihood: Consider the following regularized loss minimization:

$$\frac{1}{m}\sum_{i=1}^{m}\log(1/\mathcal{P}_\theta[x_i]) + \frac{1}{m}\left(\log(1/\theta) + \log(1/(1-\theta))\right).$$

   - Show that the preceding objective is equivalent to the usual empirical error had we added two pseudoexamples to the training set. Conclude that the regularized maximum likelihood estimator would be

$$\hat{\theta} = \frac{1}{m+2}\left(1 + \sum_{i=1}^{m} x_i\right).$$

   - Derive a high probability bound on $|\hat{\theta} - \theta^\star|$. *Hint*: Rewrite this as $|\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta^\star|$ and then use the triangle inequality and Hoeffding inequality.
   - Use this to bound the true risk. Hint: Use the fact that now $\hat{\theta} \geq \frac{1}{m+2}$ to relate $|\hat{\theta} - \theta^\star|$ to the relative entropy.
3. - Consider a general optimization problem of the form:

$$\max_{\mathbf{c}} \sum_{y=1}^{k} \nu_y \log(c_y) \quad \text{s.t.} \quad c_y > 0, \sum_y c_y = 1 \ ,$$

     where $\boldsymbol{\nu} \in \mathbb{R}_+^k$ is a vector of nonnegative weights. Verify that the M step of soft k-means involves solving such an optimization problem.
   - Let $\mathbf{c}^\star = \frac{1}{\sum_y \nu_y}\boldsymbol{\nu}$. Show that $\mathbf{c}^\star$ is a probability vector.
   - Show that the optimization problem is equivalent to the problem:

$$\min_{\mathbf{c}} D_{\mathrm{RE}}(\mathbf{c}^\star || \mathbf{c}) \quad \text{s.t.} \quad c_y > 0, \sum_y c_y = 1 \ .$$

   - Using properties of the relative entropy, conclude that $\mathbf{c}^\star$ is the solution to the optimization problem.