Chapter 14

Decision

Bayesian Decision theory. Maximum a posteriori estimation. Decisions and costs.

14.1 Bayesian Decision theory

So far, we saw several examples of maximizing likelihood as a way to find the best parameters to fit some set of labelled data. The maximum likelihood (ML) approach consists of maximizing the predicted joint probability of all features and labels, $p(x, y)$, by adjusting θ, which is the vector of parameters. Since the joint probability $p(x, y)$ can be decomposed into the conditional probability of y given x multiplied by the marginal probability$p(x)$, and since $p(x)$ is independent of the parameters of our model, θ, we can simplify the maximization problem:

$$\begin{aligned}\hat{\theta}_{ML} &= \arg\max_{\theta} \prod_{t=1}^{n} p(x^t, y^t; \theta) \\ &= \arg\max_{\theta} \prod_{t=1}^{n} p(y^t|x^t; \theta) \times \prod_{t=1}^{n} p(x^t) \\ &= \arg\max_{\theta} \prod_{t=1}^{n} p(y^t|x^t; \theta)\end{aligned} \tag{14.1}$$

In other words, we choose the parameters that maximize the probability of the observed labels given the observed features. These are the maximum likelihood parameters. Note that, in this case, the probabilities are a function of the parameters but the parameters, θ, are not considered to be random variables. This is because, under a frequentist interpretation, probability is the measure of the frequency of a random event in the limit of infinite trials and the parameters values are not random events. Thus, in a frequentist interpretation, it does not make sense to consider the probability of the parameters having those values or the probability distribution of the parameters.

However, under a Bayesian interpretation, probability is a measure of rational confidence about some value or outcome and a probability distribution describes our uncertainty about the values. Under this interpretation, θ can be considered to be just another random variable, like the features x or the labels y. Thus, under a Bayesian approach, we may want to find the most probable values of θ given the evidence obtained from the training sample S and prior assumptions regarding the probability distribution of θ. Using Bayes' rule:

$$p(\theta|S) = \frac{p(S|\theta)p(\theta)}{p(S)} \Leftrightarrow p(\theta|S) = \frac{\prod_{t=1}^{n} p(y^t|x^t,\theta)p(\theta)}{p(S)}$$

The marginal probability of the sample, the training set, $p(S)$, cannot generally be computed, but we can see it as simply a normalization value that guarantees that the probability distribution $p(\theta|S)$ integrates to 1. So, again, we can ignore this probability and find the θ that maximizes the numerator in the expression above. Thus, the *maximum a posteriori*(MAP) estimate for θ is:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \prod_{t=1}^{n} p(y^t|x^t,\theta)p(\theta) \tag{14.2}$$

In practice, the difference between equation 14.1 and equation 14.2 is that the MAP estimate includes the prior probability distribution of the parameters θ. The ML approach is equivalent to MAP when the prior probability distribution of the parameters is uniform but, if we assume a non-uniform prior distribution of θ, the results are different. This not only permits the inclusion of prior assumptions regarding reasonable values of θ but also functions as a regularization term with an explicit probabilistic justification. For example, if we assume, as a prior probability distribution for θ, that all parameter values are normally distributed with a mean of 0 and a standard deviation of 1, the MAP estimate will tend to keep the θ values close to 0 and prevent them from increasing too much, as can happen with a pure maximum likelihood estimate which does not consider θ to be a random variable or have a probability distribution.

14.2 Computing Priors

The ML approach makes no assumption about the prior probability distribution of the parameters because it does not even consider the parameters to be random variables. An *uninformative prior* is a Bayesian assumption about the prior distribution of the parameters that leads to approximately the same result as the ML approach, having no significant impact on the posterior probability. In some cases, assuming a uniform prior distribution for the model parameters θ can be an *uninformative prior*. However, sometimes this is not the case. For example, in a linear regression, the slope of the line varies from zero for a horizontal line to minus or plus infinity for a vertical line. If we assume a uniform distribution for the slope we will strongly bias our prior distribution towards nearly vertical lines, since this is where the vast majority of the values will be. Thus, in many cases, assuming a uniform distribution of the parameters is not adequate. Furthermore, there are cases where we may want to use an informative prior because we do have some information about the prior probability distribution of our parameters.

This often results in prior probability distributions for which we do not have analytical solutions for means and standard deviations. Thus, in MAP parameter estimation, it is often necessary to use numerical techniques to sample these prior probability distributions and obtain the necessary statistics. This is done by Monte Carlo techniques, most commonly by Markov Chain Monte Carlo (MCMC), which computes random walks over parameter values based on the probability distribution function in order to generate the appropriate samples. VanderPlas [22] illustrates this problem and shows some solutions using Python MCMC libraries.

In practice, even when using maximum likelihood methods in machine learning, we resort to regularization, which can be seen as a way to encourage our parameters to fall within reasonable intervals even though we do not make explicit assumptions about their prior probability distributions nor consider them to be random variables. Bayesian learning provides a more rigorous alternative, but often at significant computational cost due to the need to rely on numerical sampling methods.

14.3 Decision and Costs

Aside from the question of prior assumptions, which we can deal with explicitly in a Bayesian approach, another problem that may occur when fitting parameters to create a classifier is the cost of different mistakes.

Let us consider, as a simple example, a binary classification problem in one dimension. For each class C_1 and C_2, there is a different distribution of the feature value x, with different joint probabilities $P(x, C_1)$ and $P(x, C_2)$. If we want to create a classifier that classifies an example as C_2 if $x > \hat{x}$ or C_1, then we need to find the best value for the threshold $\hat{x}$. The probability and type of each error will depend on the value of $\hat{x}$, as illustrated in Figure 14.1. One possibility would be to minimize the total probability of committing an error, as shown on the right panel of Figure 14.1.

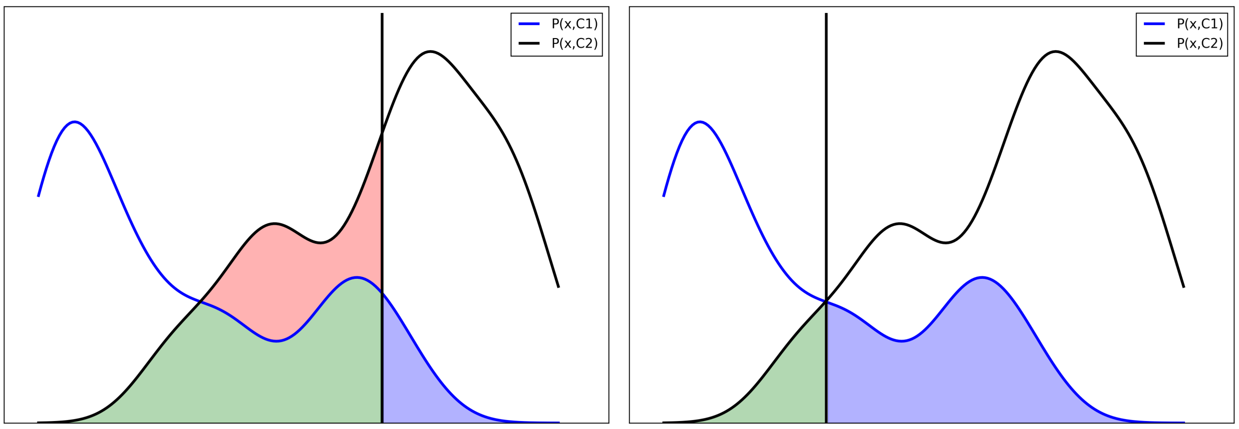

Figure 14.1: For a given value of $\hat{x}$, the threshold dividing the two classes determines the probability of each type of error. Red and green show the probability of classifying as class 1 a point of class 2 and blue the probability of classifying as class 2 a point in class 1.

However, this may not be the best option in general. Suppose class 1 corresponds to subjects with cancer and class 2 to healthy subjects. In this case, the error of misdiagnosing a healthy subject and concluding the subject has cancer is not as serious as the error of misdiagnosing a cancer patient and concluding them healthy. Let us consider the *loss matrix* shown in Table 14.1. This indicates that the cost of misdiagnosing a cancer patient is five times greater than the cost of misdiagnosing a healthy subject.

Table 14.1: Loss matrix for cancer diagnosis.

	Predict Cancer	Predict Healthy
Has Cancer	0	5
Is Healthy	1	0

Instead of trying to minimize the probability of error, we can minimize the expected loss by assigning each example to the class j that minimizes the *loss function*, which sum, for all classes k, of the joint probability of the class multiplied by the cost of classifying the example as class j when the true class is k:

$$\arg\min_j \sum_k L_{k,j} p(C_k|x)$$

In this case, the frontier dividing the two classes may not correspond to the point of lowest error probability but, rather, to the point where the error cost is minimum, as illustrated in Figure 14.2.

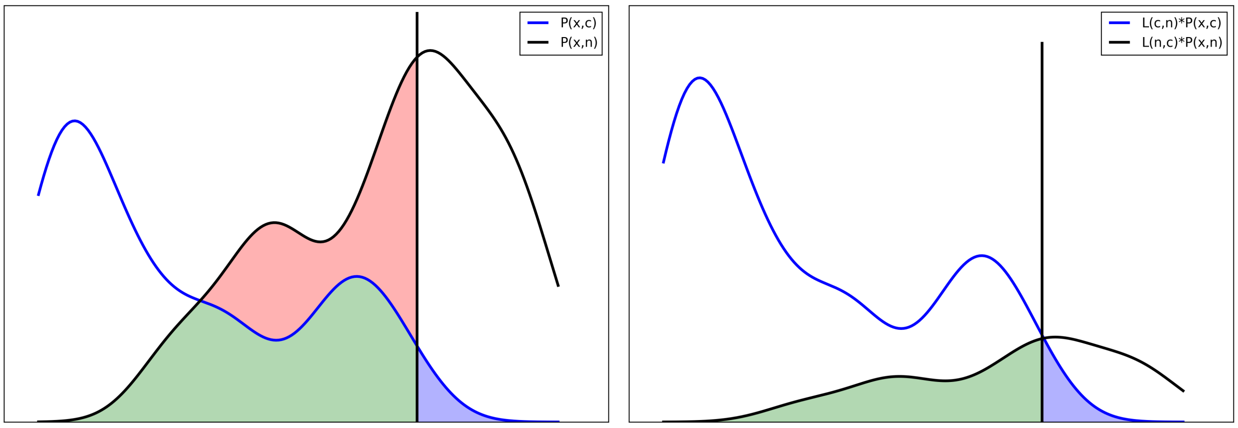

Figure 14.2: When taking into account the different costs (loss) of different errors, according to the loss matrix in Table 14.1, the frontier between the two classes gets shifter by the cost values. In this simple example, this can be understood as finding the minimum point of the probability curves multiplied by the misclassification costs (right panel).

Another problem with classification and learning is how to deal with the different levels of certainty in deciding which class or value to predict. In some cases, the predicted probability of an example belonging to one class may be only marginally larger than the probability of belonging to another class, which makes the prediction much less reliable than it would be if one probability was large and the remaining small. Figure 14.3 illustrates this problem. For low values of x, the class represented by the blue line has a much larger probability than the class represented by the black line, with the converse for high values of x. But in the mid range, the probabilities of both classes approach and a decision there is less reliable. One way to avoid this problem is to abstain from offering a classification when the probabilities for all classes, k, are below some threshold:

$$p(C_k|x) \leq \phi \qquad \forall k$$

. In the figure, this would be the 0.7 threshold. For these cases, no classification is given. This is called the *reject option*.

14.4 Further Reading

1. Alpaydin [2], Chapter 3 up to sections 3.5
2. Bishop [4], Section 1.5
3. VanderPlas, Jake, Frequentism and bayesianism: a python-driven primer, [22]

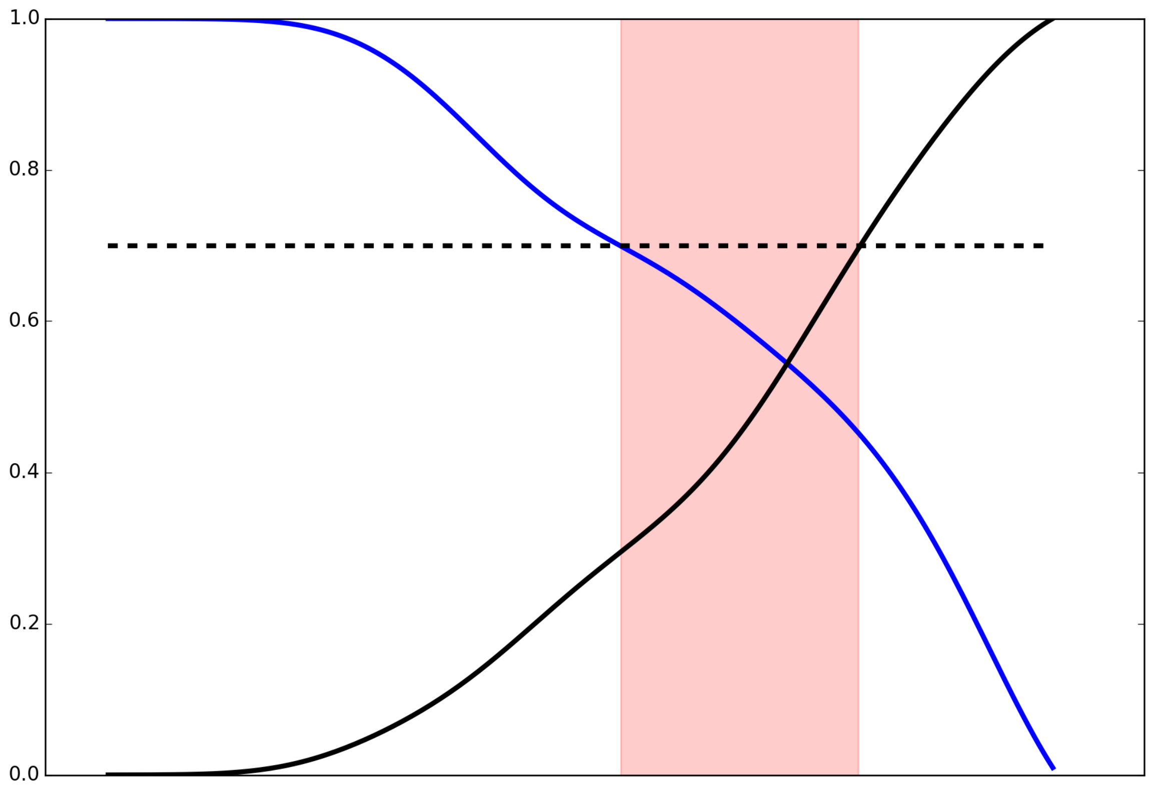

Figure 14.3: The probability of an example being in each class (blue and black) as a function of the feature value x. The region marked in red is the reject region, in which the classifier will not propose any classification because the probability for all classes is below the predefined threshold (0.7 in this case).

Bibliography

[1] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[3] David F Andrews. Plots of high-dimensional data. *Biometrics*, pages 125–136, 1972.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, New York, 1st ed. edition, oct 2006.

[5] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of www image search results using visual. Association for Computing Machinery, Inc., October 2004.

[6] Guanghua Chi, Yu Liu, and Haishandbscan Wu. Ghost cities analysis based on positioning data in china. *arXiv preprint arXiv:1510.08505*, 2015.

[7] Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[8] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning. Stanford CA Morgan Kaufmann*, pages 231–238, 2000.

[9] Hakan Erdogan, Ruhi Sarikaya, Stanley F Chen, Yuqing Gao, and Michael Picheny. Using semantic analysis to improve speech recognition performance. *Computer Speech & Language*, 19(3):321–343, 2005.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[11] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[12] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[13] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. Dna visual and analytic data mining. In *Visualization'97., Proceedings*, pages 437–441. IEEE, 1997.

[14] Chang-Hwan Lee, Fernando Gutierrez, and Dejing Dou. Calculating feature weights in naive bayes with kullback-leibler measure. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1146–1151. IEEE, 2011.

[15] Stuart Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[16] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[17] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC, 1st edition, 2009.

[18] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[19] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[20] Roberto Valenti, Nicu Sebe, Theo Gevers, and Ira Cohen. Machine learning techniques for face analysis. In Matthieu Cord and Pádraig Cunningham, editors, *Machine Learning Techniques for Multimedia*, Cognitive Technologies, pages 159–187. Springer Berlin Heidelberg, 2008.

[21] Giorgio Valentini and Thomas G Dietterich. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *The Journal of Machine Learning Research*, 5:725–775, 2004.

[22] Jake VanderPlas. Frequentism and bayesianism: a python-driven primer. *arXiv preprint arXiv:1411.5018*, 2014.