Probabilistic Machine Learning

(Chapters 13,14)[1]

David Barber

(adapted by Miguel Palomino)

[1] These slides accompany the book *Bayesian Reasoning and Machine Learning*. The book and demos can be downloaded from `www.cs.ucl.ac.uk/staff/D.Barber/brml`. Feedback and corrections are also available on the site. Feel free to adapt these slides for your own purposes, but please include a link the above website.

Supervised Learning

- Given a set of data $\mathcal{D} = \{(x^n, y^n), n = 1, \ldots, N\}$ the task is to learn the relationship between the input x and output y such that, when given a novel input x^* the predicted output y^* is accurate.
- The pair (x^*, y^*) is not in $\mathcal{D}$ but assumed to be generated by the same unknown process that generated $\mathcal{D}$.
- To specify explicitly what accuracy means one defines a loss function $L(y^{pred}, y^{true})$ or, conversely, a utility function $U = -L$.
- Our interest is describing y conditioned on knowing x, that is modelling the probability $p(y|x, \mathcal{D})$.

Unsupervised Learning

- Given a set of data $\mathcal{D} = \{x^n, n = 1, \ldots, N\}$ in unsupervised learning we aim to find a plausible compact description of the data.
- An objective is used to quantify the accuracy of the description. In unsupervised learning there is no special prediction variable so that, from a probabilistic perspective, we are interested in modelling the distribution $p(x)$.
- The likelihood of the model to generate the data is a popular measure of the accuracy of the description.

Supervised Learning: Utility and Loss

- Given a new input x^*, the optimal prediction depends on how costly making an error is.
- This can be quantified using a loss function (or conversely a utility).
- In forming a *decision function* $c(x^*)$ that will produce a class label for the new input x^*, we don't know the true class, only our surrogate for this, the predictive distribution $p(c|x^*)$.
- If $U(c^{true}, c^{pred})$ represents the utility of making a decision c^{pred} when the truth is c^{true}, the expected utility is

  $$U(c(x^*)) = \sum_{c^{true}} U(c^{true}, c(x^*))p(c^{true}|x^*)$$

  The optimal decision function $c(x^*)$ maximises the expected utility,

  $$c(x^*) = \underset{c(x^*)}{\operatorname{argmax}}\, U(c(x^*))$$

- In solving for the optimal decision function $c(x^*)$ we are assuming that the model $p(c, x)$ is correct. However, in practice we typically don't know the correct model underlying the data – all we have is a dataset of examples $\mathcal{D} = \{(x^n, c^n), n = 1, \ldots, N\}$ and our domain knowledge.

Zero-one loss/utility

A 'count the correct predictions' measure of prediction performance is based on the zero-one utility:

$$U(c^{true}, c^*) = \begin{cases} 1 \text{ if } c^* = c^{true} \\ 0 \text{ if } c^* \neq c^{true} \end{cases}$$

For the two class case, we then have the expected utility

$$U(c(x^*)) = \begin{cases} p(c^{true} = 1|x^*) \text{ for } c(x^*) = 1 \\ p(c^{true} = 2|x^*) \text{ for } c(x^*) = 2 \end{cases}$$

Hence, in order to have the highest expected utility, the decision function $c(x^*)$ should correspond to selecting the highest class probability $p(c|x^*)$:

$$c(x^*) = \begin{cases} 1 & \text{if} \quad p(c = 1|x^*) > 0.5 \\ 2 & \text{if} \quad p(c = 2|x^*) > 0.5 \end{cases}$$

In the case of a tie, either class is selected at random with equal probability.

Squared loss/utility

- In regression problems, for a real-valued prediction y^{pred} and truth y^{true}, a common loss function is the squared loss

$$L(y^{true}, y^{pred}) = \left(y^{true} - y^{pred}\right)^2$$

- The above decision framework then follows through, replacing summation with integration for the continuous variables.

Using the empirical distribution

- A direct approach to not knowing the correct model $p^{true}(c,x)$ is to replace it with the *empirical distribution*

$$p(c,x|\mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}\delta\left(c,c^n\right)\delta\left(x,x^n\right)$$

- That is, we assume that the underlying distribution is approximated by placing equal mass on each of the points (x^n,c^n) in the dataset. Using this gives the empirical expected utility

$$\left\langle U(c,c(x))\right\rangle_{p(c,x|\mathcal{D})} = \frac{1}{N}\sum_{n}U(c^n,c(x^n))$$

  or conversely the *empirical risk*

$$R = \frac{1}{N}\sum_{n}L(c^n,c(x^n))$$

- Assuming the loss is minimal when the correct class is predicted, the optimal decision $c(x)$ for any input in the train set is given by $c(x^n)=c^n$.
- However, for any new x^* not contained in $\mathcal{D}$ then $c(x^*)$ is undefined.

Empirical Risk

- In order to define the class of a novel input, one may use a parametric function $c(x|\theta)$.
- For example for a two class problem $\mathrm{dom}(c) = \{1, 2\}$, a linear decision function is given by

$$c(\mathbf{x}|\theta) = \begin{cases} 1 \text{ if } \boldsymbol{\theta}^{\mathsf{T}}\mathbf{x} + \theta_0 \geq 0 \\ 2 \text{ if } \boldsymbol{\theta}^{\mathsf{T}}\mathbf{x} + \theta_0 < 0 \end{cases}$$

If the vector input $\mathbf{x}$ is on the positive side of a hyperplane defined by the vector $\boldsymbol{\theta}$ and bias θ_0, we assign it to class 1, otherwise to class 2. The empirical risk then becomes a function of the parameters $\theta = \{\boldsymbol{\theta}, \theta_0\}$,

$$R(\theta|\mathcal{D}) = \frac{1}{N} \sum_n L(c^n, c(x^n|\theta))$$

The optimal parameters θ are given by minimising the empirical risk with respect to θ,

$$\theta_{opt} = \underset{\theta}{\operatorname{argmin}}\ R(\theta|\mathcal{D})$$

The decision for a new datapoint x^* is then given by $c(x^*|\theta_{opt})$.

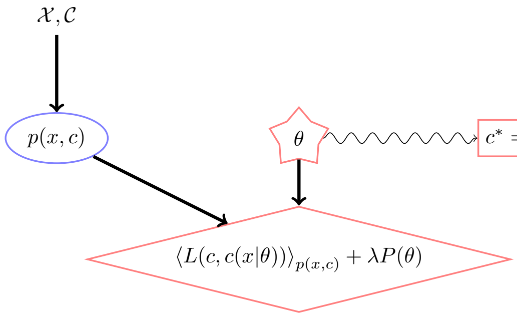
Empirical Risk

- In this *empirical risk minimisation* approach, as we make the decision function $c(x|\theta)$ more flexible, the empirical risk goes down.
- However, if we make $c(x|\theta)$ too flexible we will have little confidence that $c(x|\theta)$ will perform well on a novel input x^*. The reason for this is that a flexible decision function $c(x|\theta)$ is one for which the class label can change for only a small change in x. Such flexibility seems good since it means that we will be able to find a parameter setting θ so that the train data is fitted well.
- However, since we only constrain the decision function on the known training points, a flexible $c(x|\theta)$ may change rapidly as we move away from the train data, leading to poor generalisation.
- To constrain the complexity of $c(x|\theta)$ we may minimise the penalised empirical risk

$$R'(\theta|\mathcal{D}) = R(\theta|\mathcal{D}) + \lambda P(\theta)$$

where $P(\theta)$ is a function that penalises complex functions $c(x|\theta)$. The *regularisation constant,* λ, determines the strength of this penalty and is typically set by validation.

Empirical Risk


$\mathcal{X}, \mathcal{C}$
x^*
$p(x,c)$
θ
$c^* = c(x^*|\theta)$
$\langle L(c, c(x|\theta))\rangle_{p(x,c)} + \lambda P(\theta)$

Empirical risk approach. Given the dataset $\mathcal{X}, \mathcal{C}$, a model of the data $p(x,c)$ is made, usually using the empirical distribution. For a classifier $c(x|\theta)$, the parameter θ is learned by minimising the penalised empirical risk with respect to θ. The penalty parameter λ is set by validation. A novel input x^* is then assigned to class $c(x^*|\theta)$, given this optimal θ.

Benefits of the empirical risk approach

- In the limit of a large amount of training data the empirical distribution tends to the correct distribution.
- The discriminant function is chosen on the basis of minimal risk, which is the quantity we are ultimately interested in.
- The procedure is conceptually straightforward.

Drawbacks of the empirical risk approach

- It seems extreme to assume that the data follows the empirical distribution, particularly for small amounts of training data. More reasonable assumptions for $p(x)$ would take into account likely x that could arise, not just those in the train data.
- If the loss function changes, the discriminant function needs to be retrained.
- Some problems require an estimate of the confidence of the prediction. Whilst there may be heuristic ways to evaluating confidence in the prediction, this is not inherent in the framework.
- When there are many penalty parameters, performing cross-validation in a discretised grid of the parameters becomes infeasible.
- During validation, many models are trained, and all but one subsequently discarded.

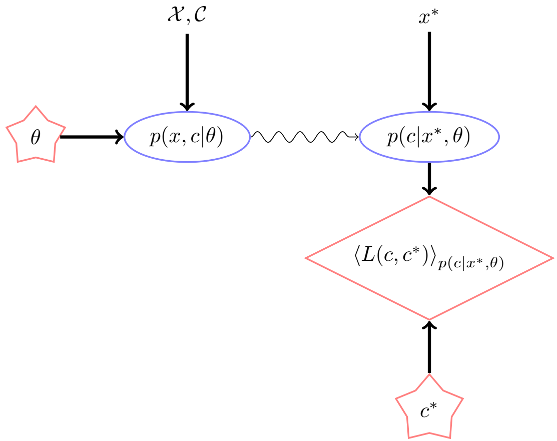
Bayesian decision approach

- An alternative to using the empirical distribution is to first fit a model $p(c, x|\theta)$ to the train data $\mathcal{D}$.
- Given this model, the decision function $c(x)$ is automatically determined from the maximal expected utility (or minimal risk) with respect to this model, in which the unknown $p(c^{true}|x)$ is replaced with $p(c|x, \theta)$.
- There are two main approaches to fitting $p(c, x|\theta)$ to data $\mathcal{D}$. We could parameterise the joint distribution using

$$p(c, x|\theta) = p(c|x, \theta_{c|x})p(x|\theta_x) \qquad \text{discriminative approach}$$

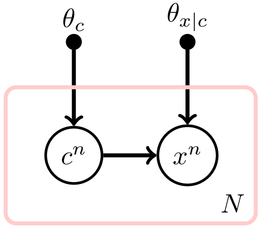
or

$$p(c, x|\theta) = p(x|c, \theta_{x|c})p(c|\theta_c) \qquad \text{generative approach}$$

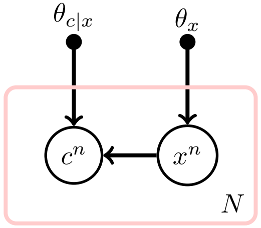

$\mathcal{X}, \mathcal{C}$
x^*
θ
$p(x, c|\theta)$
$p(c|x^*, \theta)$
$\langle L(c, c^*)\rangle_{p(c|x^*,\theta)}$
c^*

Bayesian decision approach. A model $p(x, c|\theta)$ is fitted to the data. After learning the optimal model parameters θ, we compute $p(c|x, \theta)$. For a novel x^*, the distribution of the assumed 'truth' is $p(c|x^*, \theta)$. The prediction (decision) is then given by that c^* which minimises the expected risk $\langle L(c, c^*)\rangle_{p(c|x^*,\theta)}$.

Two approaches to specifying the model



(a)



(b)

Two generic strategies for probabilistic classification. **(a)**: Class dependent generative model of x. After learning parameters, classification is obtained by making x evidential and inferring $p(c|x)$. **(b)**: A discriminative classification method $p(c|x)$.

Generative approach

Advantages: Prior information about the structure of the data is often most naturally specified through a generative model $p(x|c)$. For example, for male faces, we would expect to see heavier eyebrows, a squarer jaw, *etc.*

Disadvantages: The generative approach does not directly target the classification model $p(c|x)$ since the goal of generative training is rather to model $p(x|c)$. If the data x is complex, finding a suitable generative data model $p(x|c)$ is a difficult task. Furthermore, since each generative model is separately trained for each class, there is no competition amongst the models to explain the x data. On the other hand it might be that making a model of $p(c|x)$ is simpler, particularly if the decision boundary between the classes has a simple form, even if the data distribution of each class is complex.

Discriminative approach

Advantages The discriminative approach directly addresses finding an accurate classifier $p(c|x)$ based on modelling the decision boundary, as opposed to the class conditional data distribution in the generative approach. Whilst the data from each class may be distributed in a complex way, it could be that the decision boundary between them is relatively easy to model.

Disadvantages Discriminative approaches are usually trained as 'black-box' classifiers, with little prior knowledge built used to describe how data for a given class is distributed. Domain knowledge is often more easily expressed using the generative framework.

Benefits of the Bayesian decision approach

- This is a conceptually ‘clean’ approach, in which one tries one’s best to model the environment (using either a generative or discriminative approach), independent of the subsequent decision process.
- In this case learning the environment is separated from the effect this will have on the expected utility.
- If $p(x, c|\theta)$ is the ‘true’ model of the data, this approach is optimal.

Drawbacks of the Bayesian decision approach

- If the environment model $p(c, x|\theta)$ is poor, the prediction c^* could be highly inaccurate since modelling the environment is divorced from prediction.
- To avoid fully divorcing the learning of the model $p(c, x|\theta)$ from its effect on decisions, in practice one often includes regularisation terms in the environment model $p(c, x|\theta)$ which are set by validation based on an empirical loss.

Example: K-Nearest Neighbours

- Each input vector $\mathbf{x}$ has a corresponding class label, $c^n \in \{1, \ldots, C\}$. Given a dataset of N train examples, $\mathcal{D} = \{\mathbf{x}^n, c^n\}, n = 1, \ldots, N$, and a novel $\mathbf{x}$, we aim to return the correct class $c(\mathbf{x})$.
- Successful prediction typically relies on smoothness in the data.
- Nearest neighbour methods are a useful starting point since they readily encode basic smoothness intuitions and are easy to program.
- For novel $\mathbf{x}$, find the nearest input in the training set and use the class of this nearest input.
- If your neighbour is simply mistaken (has an incorrect training class label), or is not a particularly representative example of his class, then these situations will typically result in an incorrect classification.
- By including more than the single nearest neighbour, we hope to make a more robust classifier with a smoother decision boundary (less swayed by single neighbour opinions).

Probabilistic Interpretation of Nearest Neighbours

Consider the situation where we have data from two classes – class 0 and class 1. We make the following mixture model for data from class 0, placing a Gaussian on each datapoint:

$$p(\mathbf{x}|c=0) = \frac{1}{N_0} \sum_{n \in \text{ class } 0} \mathcal{N}\left(\mathbf{x}|\mathbf{x}^n, \sigma^2 \mathbf{I}\right)$$

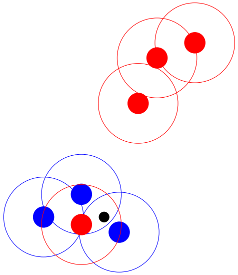
where D is the dimension of a datapoint $\mathbf{x}$ and N_0 are the number of train points of class 0, and σ^2 is the variance.
Similarly, for data from class 1:

$$p(\mathbf{x}|c=1) = \frac{1}{N_1} \sum_{n \in \text{ class } 1} \mathcal{N}\left(\mathbf{x}|\mathbf{x}^n, \sigma^2 \mathbf{I}\right)$$

To classify a new datapoint $\mathbf{x}^*$, we use Bayes' rule

$$p(c=0|\mathbf{x}^*) = \frac{p(\mathbf{x}^*|c=0)p(c=0)}{p(\mathbf{x}^*|c=0)p(c=0) + p(\mathbf{x}^*|c=1)p(c=1)}$$

Probabilistic Interpretation



A probabilistic interpretation of nearest neighbours. For each class we use a mixture of Gaussians to model the data from that class $p(\mathbf{x}|c)$, placing at each training point an isotropic Gaussian of width σ^2. The width of each Gaussian is represented by the circle. In the limit $\sigma^2 \to 0$ a novel point (small black dot) is assigned the class of its nearest neighbour. For finite $\sigma^2 > 0$ the influence of non-nearest neighbours has an effect, resulting in a soft version of nearest neighbours.

Maximum Likelihood

- The maximum likelihood setting of $p(c = 0)$ is $N_0/(N_0 + N_1)$, and $p(c = 1) = N_1/(N_0 + N_1)$.
- To see which class is most likely we may use the ratio

$$\frac{p(c = 0|\mathbf{x}^*)}{p(c = 1|\mathbf{x}^*)} = \frac{p(\mathbf{x}^*|c = 0)p(c = 0)}{p(\mathbf{x}^*|c = 1)p(c = 1)} \tag{1}$$

If this ratio is greater than one, we classify $\mathbf{x}^*$ as 0, otherwise 1.

Limiting case

- If σ^2 is very small, the numerator is dominated by that term for which datapoint $\mathbf{x}^{n_0}$ in class 0 is closest to the point $\mathbf{x}^*$. Similarly, the denominator will be dominated by that datapoint $\mathbf{x}^{n_1}$ in class 1 which is closest to $\mathbf{x}^*$. This is because, when $x < y$

$$e^{-x/\sigma} + e^{-y/\sigma} = e^{-x/\sigma}(1 + e^{x-y/\sigma}) \approx e^{-x/\sigma}$$

and the second factor tends to 1 when σ tends to 0: Hence:

$$\frac{p(c=0|\mathbf{x}^*)}{p(c=1|\mathbf{x}^*)} \approx \frac{e^{-(\mathbf{x}^*-\mathbf{x}^{n_0})^2/(2\sigma^2)}}{e^{-(\mathbf{x}^*-\mathbf{x}^{n_1})^2/(2\sigma^2)}}$$

- Taking the limit $\sigma^2 \to 0$, with certainty we classify $\mathbf{x}^*$ as class 0 if $\mathbf{x}^*$ is closer to $\mathbf{x}^{n_0}$ than to $\mathbf{x}^{n_1}$.
- The nearest (single) neighbour method is therefore recovered as the limiting case of a probabilistic generative model.

Probabilistic Interpretation

- The motivation for K nearest neighbours is to produce a classification that is robust against unrepresentative single nearest neighbours.
- To ensure a similar kind of robustness in the probabilistic interpretation, we may use a finite value $\sigma^2 > 0$.
- This smoothes the extreme probabilities of classification and means that more points (not just the nearest) will have an effective contribution in (1).
- The extension to more than two classes is straightforward, requiring a class conditional generative model for each class.