Supervised Learning

Prof. Dr. H. H. Takada

Quantitative Research – Itaú Asset Management
Institute of Mathematics and Statistics – University of São Paulo

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 $c(x^*)$ maximizes 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.

General utility functions

- In general, for a two-class problem, we have

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

  and the optimal decision function $c(x^*)$ chooses that class with highest expected utility.

- One can readily generalize this to multiple-class situations using a *utility matrix* with elements

$$U_{ij} = U(c^{true} = i, c^{pred} = j)$$

  where the i, j element of the matrix contains the utility of predicting class j when the true class is i.

Asymmetric Utility

- In some applications the utility matrix is highly non-symmetric.
- Consider a medical scenario in which we are asked to predict whether or not the patient has cancer $\mathrm{dom}(c) = \{\text{cancer}, \text{benign}\}$.
- If the true class is cancer yet we predict benign, this could have terrible consequences for the patient. On the other hand, if the class is benign yet we predict cancer, this may be less disastrous for the patient.
- Such asymmetric utilities can favour conservative decisions – in the cancer case, we would be more inclined to decide the sample is cancerous than benign, even if the predictive probability of the two classes is equal.

Squared loss/utilty

- 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.
- For an output prediction given an input x,

$$L(y^{pred}) = \int \left(y^{true} - y^{pred}\right)^2 p(y^{true}|x) dy^{true}$$

This loss is minimized by setting

$$y^{pred} = \int y^{true} p(y^{true}|x) dy^{true}$$

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

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

  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 minimizing the empirical risk with respect to θ,

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

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

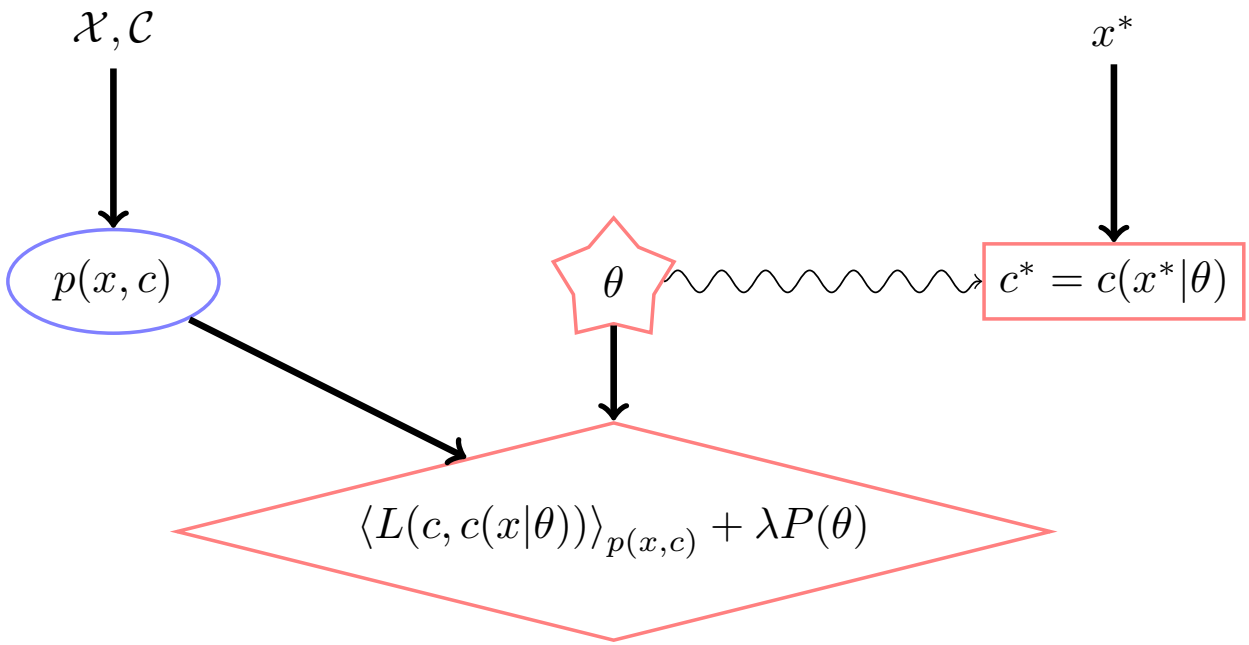
Empirical Risk

- In this *empirical risk minimization* 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 generalization.
- To constrain the complexity of $c(x|\theta)$ we may minimize the penalized empirical risk

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

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

Empirical Risk

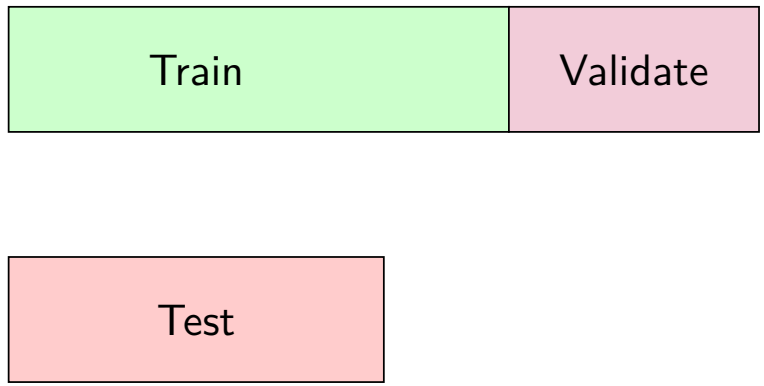


Empirical risk approach. Given the dataset $\mathcal{X}$ and $\mathcal{C}$, a model of the data $p(x, c)$ is made, usually using the empirical distribution. For a classifier $c(x|\theta)$, the parameter θ is learned by minimizing the penalized empirical risk with respect to θ. The penalty parameter λ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

Heuristic justification for squared Penalty

- For the linear decision function above, it is reasonable to penalize wildly changing classifications in the sense that if we change the input $\mathbf{x}$ by only a small amount we expect (on average) minimal change in the class label.
- The squared difference in $\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}+\theta_0$ for two inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ is $\left(\boldsymbol{\theta}^{\mathsf{T}}\Delta\mathbf{x}\right)^2$ where $\Delta\mathbf{x} \equiv \mathbf{x}_2 - \mathbf{x}_1$.
- By constraining the length of $\boldsymbol{\theta}$ to be small we limit the ability of the classifier to change class for only a small change in the input space.
- Assuming the distance between two datapoints is distributed according to an isotropic multivariate Gaussian with zero mean and covariance $\sigma^2\mathbf{I}$, the average squared change is $\left\langle \left(\boldsymbol{\theta}^{\mathsf{T}}\Delta\mathbf{x}\right)^2 \right\rangle = \sigma^2\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\theta}$, motivating the choice of the Euclidean squared length of the parameter $\boldsymbol{\theta}$ as the penalty term, $P(\theta) = \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\theta}$.

Validation

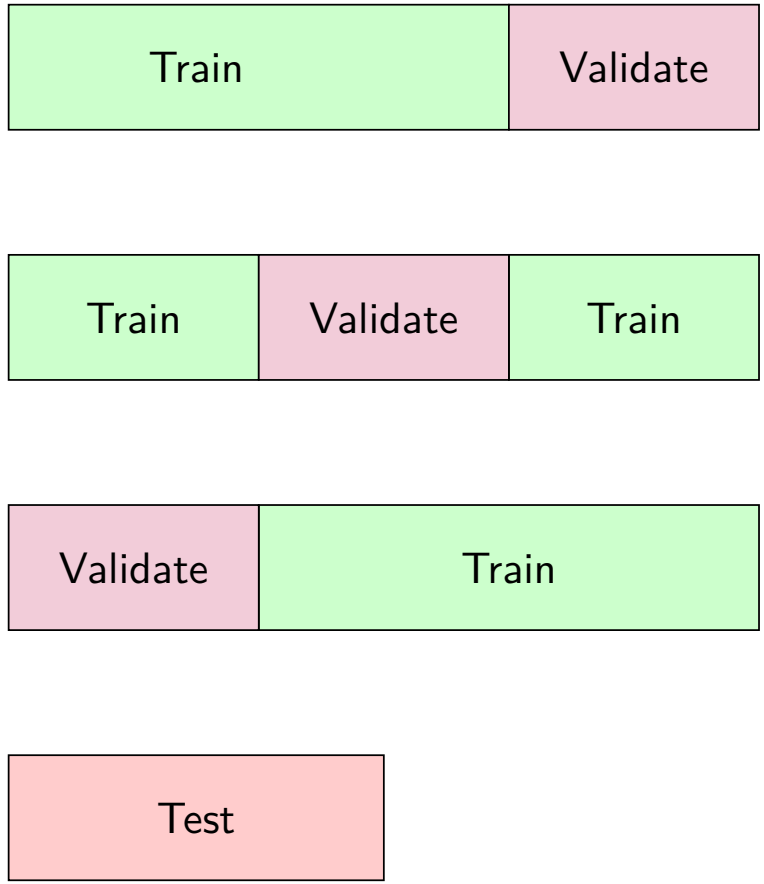


Models can be trained using the train data based on different regularization parameters. The optimal regularization parameter is determined by the empirical performance on the validation data. An independent measure of the generalization performance is obtained by using a separate test set.

Validation

- In penalized empirical risk minimization we need to set the regularization constant λ. This can be achieved by evaluating the performance of the learned classifier $c(x|\theta)$ on validation data $\mathcal{D}_{validate}$ for several different λ values, and choosing the λ which gave rise to the classifier with the best performance.
- It's important that the validation data is not the data on which the model was trained since we know that the optimal setting for λ in that case is zero, and again we will have little confidence in the generalization ability.
- Given a dataset $\mathcal{D}$ we split this into disjoint parts, $\mathcal{D}_{train}, \mathcal{D}_{validate}$, where the size of the validation set is usually chosen to be smaller than the train set. For each parameter λ_a one then finds the minimal empirical risk parameter θ_a. The optimal λ is chosen as that which gives rise to the model with the minimal validation risk. Using the optimal regularization parameter λ, many practitioners retrain θ on the basis of the whole dataset $\mathcal{D}$.

Cross-Validation

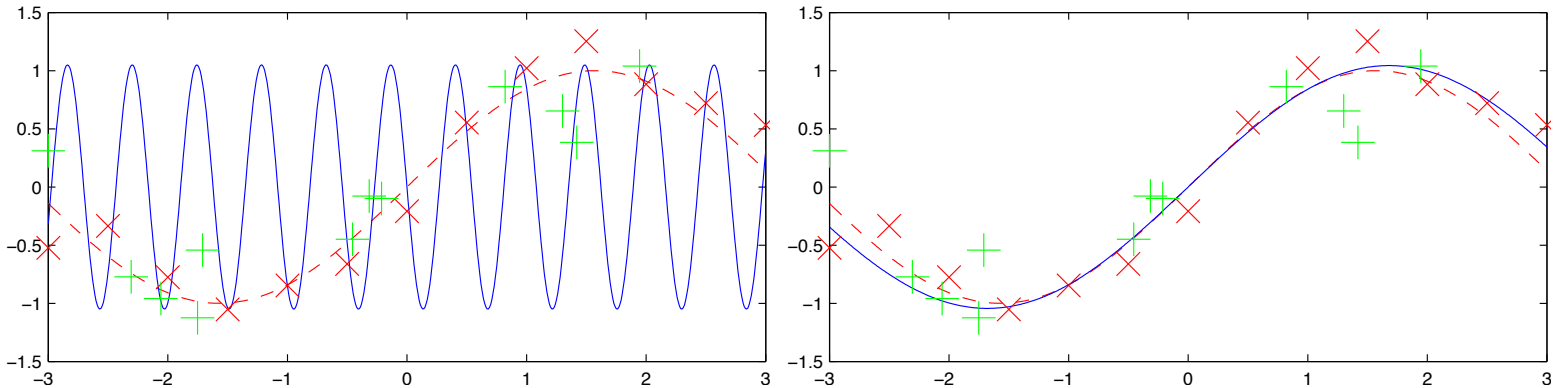


In cross-validation the dataset is split into several train-validation sets. Depicted is 3-fold cross-validation. For a range of regularization parameters, the optimal regularization parameter is found based on the empirical validation performance averaged across the different splits.

Cross-Validation

- In cross-validation the dataset is partitioned into training and validation sets multiple times with validation results obtained for each partition.
- Each partition produces a different training $\mathcal{D}^i_{train}$ and validation $\mathcal{D}^i_{validate}$ set, along with an optimal penalized empirical risk parameter θ^i_a and associated (unregularized) validation performance $R(\theta^i_a|\mathcal{D}^i_{validate})$.
- The performance of regularization parameter λ_a is taken as the average of the validation performances over i. The best regularization parameter is then given as that with the minimal average validation error.
- In K-fold cross-validation the data $\mathcal{D}$ is split into K equal sized disjoint parts $\mathcal{D}_1, \ldots, \mathcal{D}_K$. Then $\mathcal{D}^i_{validate} = \mathcal{D}_i$ and $\mathcal{D}^i_{train} = \mathcal{D} \backslash \mathcal{D}^i_{validate}$. This gives a total of K different training-validation sets over which performance is averaged. In practice 10-fold cross-validation is popular, as is leave-one-out cross-validation in which the validation sets consist of only a single example.

Validation example



(a) (b)

The true function which generated the noisy data is the dashed line; the function learned from the data is given by the solid line. **(a)**: The unregularised fit ($\lambda = 0$) to training given by $\times$. Whilst the training data is well fitted, the error on the validation examples, denoted by $+$, is high. **(b)**: The regularised fit ($\lambda = 0.5$). Whilst the train error is high, the validation error is low.

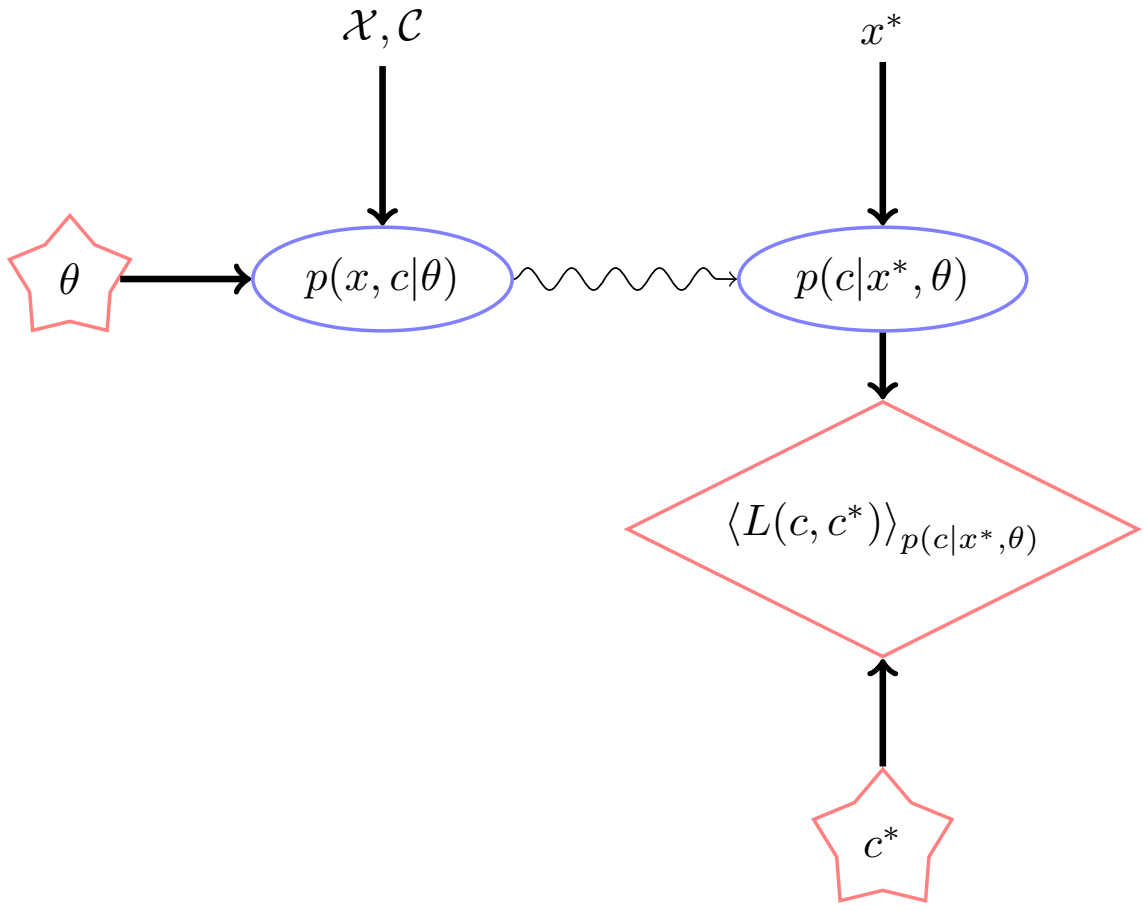
Bayesian decision approach

- 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 parameterize the joint distribution using

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

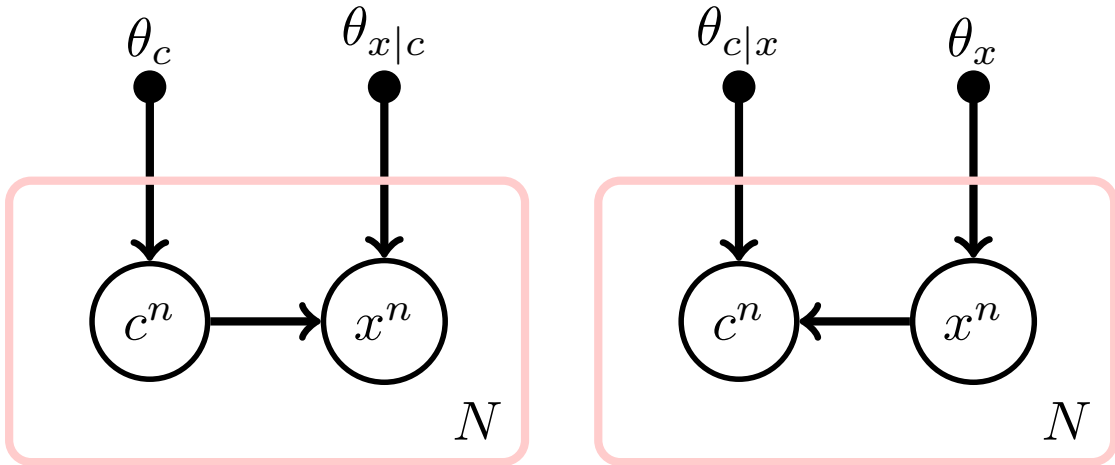
or

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



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 minimizes the expected risk $\langle L(c, c^*) \rangle_{p(c|x^*,\theta)}$.

Two approaches to specifying the model



(c) (d)

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

Generative approach $p(\mathbf{x}, c|\theta) = p(\mathbf{x}|c, \theta_{x|c})p(c|\theta_c)$

For simplicity we use maximum likelihood training for the parameters θ. Assuming the data $\mathcal{D}$ is i.i.d., we have a loglikelihood

$$\log p(\mathcal{D}|\theta) = \sum_n \log p(\mathbf{x}^n|c^n, \theta_{x|c}) + \sum_n \log p(c^n|\theta_c)$$

As we see, the dependence on $\theta_{x|c}$ occurs only in the first term, and θ_c only occurs in the second.

Example: Male-Female Images. This means that learning the optimal parameters is equivalent to isolating the data for the male-class and fitting a model $p(\mathbf{x}|c = \text{male}, \theta_{x|\text{male}})$. We may similarly isolate the female data and fit a separate model $p(\mathbf{x}|c = \text{female}, \theta_{x|\text{female}})$. The class distribution $p(c|\theta_c)$ is set according to the ratio of males/females in the set of training data.

To make a classification of a new image $\mathbf{x}^*$ as either male or female, we use Bayes' rule:

$$p(c = \text{male}|\mathbf{x}^*) = \frac{p(\mathbf{x}^*, c = \text{male}|\theta_{x|\text{male}})}{p(\mathbf{x}^*, c = \text{male}|\theta_{x|\text{male}}) + p(\mathbf{x}^*, c = \text{female}|\theta_{x|\text{female}})} \quad (1)$$

Based on zero-one loss, if this probability is greater than 0.5 we classify $\mathbf{x}^*$ as male, otherwise female. For a general loss function, we use this probability as part of a decision process.

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 $p(\mathbf{x}, c|\theta) = p(c|\mathbf{x}, \theta_{c|x})p(\mathbf{x}|\theta_x)$

Assuming i.i.d. data, the log likelihood is

$$\log p(\mathcal{D}|\theta) = \sum_n \log p(c^n|\mathbf{x}^n, \theta_{c|x}) + \sum_n \log p(\mathbf{x}^n|\theta_x)$$

The parameters are isolated in the two terms so that maximum likelihood training is equivalent to finding the parameters of $\theta_{c|x}$ that will best predict the class c for a given training input x. The parameters θ_x for modelling the data occur only in the second term above, and setting them can therefore be treated as a separate unsupervised learning problem. This approach consequently isolates modelling the *decision boundary* from modelling the input distribution. Classification of a new point $\mathbf{x}^*$ is based on

$$p(c|\mathbf{x}, \theta^{opt}_{c|x})$$

As for the generative case, this approach still learns a joint distribution $p(c, x) = p(c|x)p(x)$ which can be used as part of a decision process if required.

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.

Hybrid generative-discriminative approaches

One could use a generative description, $p(x|c)$, building in prior information, and use this to form a joint distribution $p(x, c)$, from which a discriminative model $p(c|x)$ may be formed, using Bayes' rule. Specifically, we can use

$$p(c|x, \theta) = \frac{p(x|c, \theta_{x|c})p(c|\theta_c)}{\sum_c p(x|c, \theta_{x|c})p(c|\theta_c)}$$

Subsequently the parameters $\theta = \left(\theta_{x|c}, \theta_c\right)$, for this hybrid model can be found by maximizing the probability of being in the correct class. A separate model is learned for $p(x|\theta_x)$. This approach would appear to leverage the advantages of both the discriminative and generative frameworks since we can more readily incorporate domain knowledge in the generative model $p(x|c, \theta_{x|c})$ yet train this in a discriminative way. This approach is rarely taken in practice since the resulting functional form of the likelihood depends in a complex manner on the parameters. In this case no parameter separation between θ_c and $\theta_{x|c}$ occurs (as was previously the case for the generative and discriminative approaches).

Features and preprocessing

It is often the case that in discriminative training, transforming the raw input x into a form that more directly captures the relevant label information can greatly improve performance. For example, in the male-female classification case, it might be that building a classifier directly in terms of the elements of the face vector $\mathbf{x}$ is difficult. However, using 'features' which contain geometric information such as the distance between eyes, width of mouth, *etc.* may make finding a classifier easier. In practice data is also often preprocessed to remove noise, centre an image *etc.*

Learning lower-dimensional representations in semi-supervised learning

One way to exploit a large amount of unlabelled training data to improve classification is to first find a lower dimensional representation h of the data x. Based on this, the mapping from h to c may be rather simpler to learn than a mapping from x to c directly. We use $c^n = \emptyset$ to indicate that the class for datapoint n is missing. We can then form the likelihood on the visible data using,

$$p(\mathcal{C}, \mathcal{X}, \mathcal{H}|\theta) = \prod_n \left\{p(c^n|h^n, \theta_{c|h})\right\}^{\mathbb{I}[c^n \neq \emptyset]} p(x^n|h^n, \theta_{x|h}) p(h|\theta_h)$$

and set any parameters for example by using maximum likelihood

$$\theta^{opt} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathcal{H}} p(\mathcal{C}, \mathcal{X}, \mathcal{H}|\theta)$$

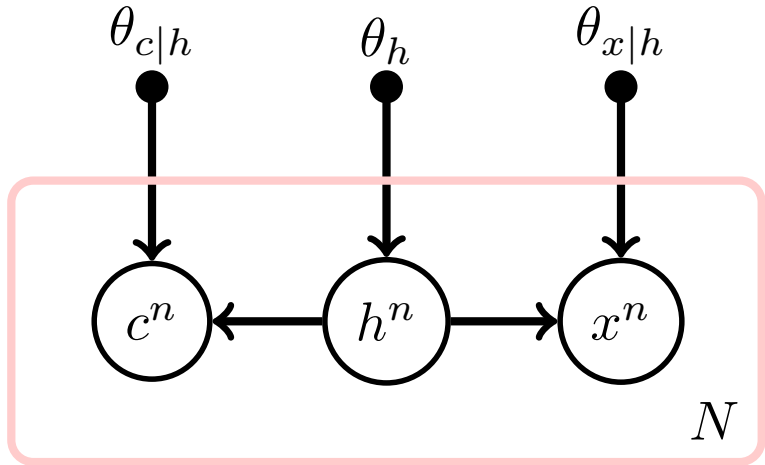
Benefits of the Bayesian decision approach

- This is a conceptually 'clean' approach, in which one tries ones 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.
- The decision c^* for a novel input x^* can be a highly complex function of x^* due to the maximization operation.
- 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 regularization terms in the environment model $p(c, x|\theta)$ which are set by validation based on an empirical loss.

Semisupervised learning



A strategy for semi-supervised learning. When c^n is missing, the term $p(c^n|h^n)$ is absent. The large amount of training data helps the model learn a good lower dimension/compressed representation h of the data x. Fitting then a classification model $p(c|h)$ using this lower dimensional representation may be much easier than fitting a model directly from the complex data to the class, $p(c|x)$.

Bayes versus Empirical Decisions

The empirical risk and Bayesian approaches are at the extremes of the philosophical spectrum. In the empirical risk approach one makes a seemingly over-simplistic data generating assumption. However decision function parameters are set based on the task of making decisions. On the other hand, the Bayesian approach attempts to learn meaningful $p(c, x)$ without regard to its ultimate use as part of a larger decision process. What 'objective' criterion can we use to learn $p(c, x)$, particularly if we are only interested in classification with a low test-risk? The following example is intended to recapitulate the two generic Bayes and empirical risk approaches we've been considering. Note that we've previously written utilities suggesting that they are both for example class labels; however the theory applies more generally, for example to utilities $u(c, d)$ where d is some decision (which is not necessarily of the form of the class label c).

Bayes versus Empirical Decisions

The two generic decision strategies

Consider a situation in which, based on patient information $\mathbf{x}$, we need to take a decision d as whether or not to operate. The utility of operating $u(c, d)$ depends on whether or not the patient has cancer, c. For example

$$u(\text{cancer}, \text{operate}) = 100 \qquad u(\text{benign}, \text{operate}) = 30$$
$$u(\text{cancer}, \text{don't operate}) = 0 \quad u(\text{benign}, \text{don't operate}) = 70$$

We have independent true assessments of whether or not a patient had cancer, giving rise to a set of historical records $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$. Faced with a new patient with information $\mathbf{x}$, we need to make a decision whether or not to operate.

Bayes versus Empirical Decisions

In the Bayesian decision approach one would first make a model $p(c|\mathbf{x}, \mathcal{D})$ (for example using a discriminative model such as logistic regression). Using this model the decision is given by that which maximizes the expected utility

$$d = \underset{d}{\text{argmax}} \ [p(\text{cancer}|\mathbf{x}, \mathcal{D})u(\text{cancer}, d) + p(\text{benign}|\mathbf{x}, \mathcal{D})u(\text{benign}, d)]$$

In this approach learning the model $p(c|\mathbf{x}, \mathcal{D})$ is divorced from the ultimate use of the model in the decision making process. An advantage of this approach is that, from the viewpoint of expected utility, it is optimal – provided the model $p(c|\mathbf{x}, \mathcal{D})$ is 'correct'. Unfortunately, this is rarely the case. Given the limited model resources, it might make sense to focus on ensuring the prediction of cancer is correct since this has a more significant effect on the utility. However, formally, this would require a corruption of this framework.

Bayes versus Empirical Decisions

The alternative empirical utility approach recognizes that the task can be stated as to translate patient information $\mathbf{x}$ into an operation decision d. To do so one could parameterize this as $d(\mathbf{x}) = f(\mathbf{x}|\boldsymbol{\theta})$ and then learn $\boldsymbol{\theta}$ under maximizing the empirical utility

$$u(\boldsymbol{\theta}) = \sum_n u(f(x^n|\boldsymbol{\theta}), c^n)$$

For example, if $\mathbf{x}$ is a vector representing the patient information and $\boldsymbol{\theta}$ the parameter, we might use a linear decision function such as

$$f(\mathbf{x}|\boldsymbol{\theta}) = \begin{cases} \boldsymbol{\theta}^\mathsf{T}\mathbf{x} \geq 0 & d = \text{operate} \\ \boldsymbol{\theta}^\mathsf{T}\mathbf{x} < 0 & d = \text{don't operate} \end{cases}$$

The advantage of this approach is that the parameters of the decision are directly related to the utility of making the decision. However, it may be that we have a good model of $p(c|\mathbf{x})$ and would wish to make use of this. A disadvantage is that we cannot easily incorporate such domain knowledge into the decision function.

Bayes versus Empirical Decisions

Both approaches are heavily used in practice and which is to be preferred depends very much on the problem. Whilst the Bayesian approach appears formally optimal, it is prone to model mis-specification. A pragmatic alternative Bayesian approach is to fit a parameterized distribution $p(c, x|\lambda)$ to the data $\mathcal{D}$, where λ penalizes complexity of the fitted distribution, setting λ using validation on the risk. This has the potential advantage of allowing one to incorporate sensible prior information about $p(c, x)$ whilst assessing competing models in the light of their actual predictive risk. Similarly, for the empirical risk approach, one can modify the extreme empirical distribution assumption by using a more plausible model $p(c, x)$ of the data.