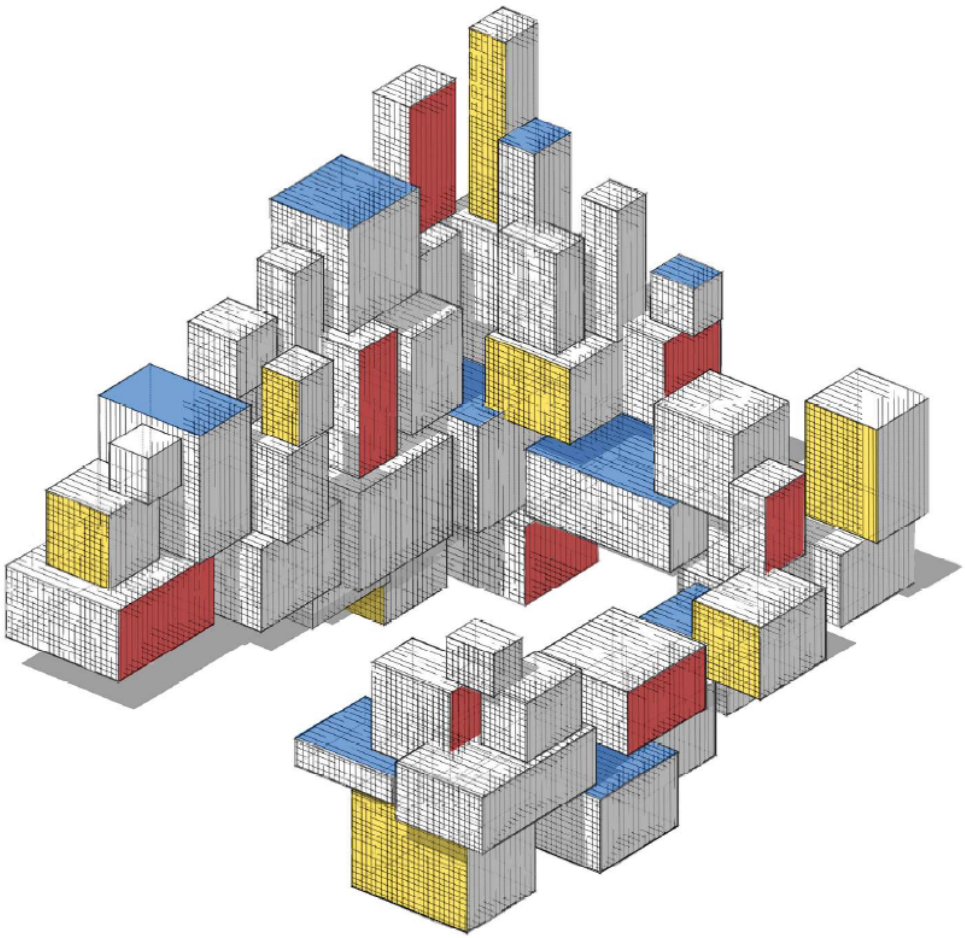
Classification



Purpose

In this lecture we discuss:

- The general classification framework
- Zero-one loss
- Bayes error rate
- Classification metrics

Classification Methods

These are supervised learning methods in which a categorical response variable Y takes one of c possible values (for example whether a person is sick or healthy), which is to be predicted from a vector $\boldsymbol{X}$ of explanatory variables (for example, the blood pressure, age, and smoking status of the person), using a prediction function g.

In this sense, g classifies the input $\boldsymbol{X}$ into one of the classes, say in the set $\{0, \ldots, c-1\}$. For this reason, we will call g a classification function or simply classifier.

As with any supervised learning technique, the goal is to minimize the expected loss or risk

$$\ell(g) = \mathbb{E}\,\mathrm{Loss}(Y, g(\boldsymbol{X})) \tag{1}$$

for some loss function, $\mathrm{Loss}(y, \widehat{y})$, that quantifies the impact of classifying a response y via $\widehat{y} = g(\boldsymbol{x})$.

Zero-one Loss

The natural loss function is the zero–one (also written 0–1) or indicator loss: $\text{Loss}(y, \widehat{y}) := \mathbb{I}\{y \neq \widehat{y}\}$; that is, there is no loss for a correct classification ($y = \widehat{y}$) and a unit loss for a misclassification ($y \neq \widehat{y}$).

Theorem: Optimal classifier

For the loss function $\text{Loss}(y, \widehat{y}) = \mathbb{I}\{y \neq \widehat{y}\}$, an optimal classification function is

$$g^*(\boldsymbol{x}) = \underset{y \in \{0,\ldots,c-1\}}{\text{argmax}} \; \mathbb{P}[Y = y \mid \boldsymbol{X} = \boldsymbol{x}]. \tag{2}$$

Proof

The goal is to minimize $\ell(g) = \mathbb{E}\,\mathbb{I}\{Y \neq g(\boldsymbol{X})\}$ over all functions g taking values in $\{0, \ldots, c-1\}$.

Conditioning on $\boldsymbol{X}$ gives, by the tower property, $\ell(g) = \mathbb{E}\left(\mathbb{P}[Y \neq g(\boldsymbol{X}) \mid \boldsymbol{X}]\right)$, and so minimizing $\ell(g)$ with respect to g can be accomplished by maximizing $\mathbb{P}[Y = g(\boldsymbol{x}) \mid \boldsymbol{X} = \boldsymbol{x}]$ with respect to $g(\boldsymbol{x})$, for every fixed $\boldsymbol{x}$.

In other words, take $g(\boldsymbol{x})$ to be equal to the class label y for which $\mathbb{P}[Y = y \mid \boldsymbol{X} = \boldsymbol{x}]$ is maximal. □

Bayes Error Rate

The formulation (2) allows for "ties", when there is an equal probability between optimal classes for a feature vector $\boldsymbol{x}$.

The optimal prediction function depends on the conditional pdf $f(y \mid \boldsymbol{x}) = \mathbb{P}[Y = y \mid \boldsymbol{X} = \boldsymbol{x}]$.

Since we assign $\boldsymbol{x}$ to class y if $f(y \mid \boldsymbol{x}) \geqslant f(z \mid \boldsymbol{x})$ for all z, we do not need to learn the entire surface of the function $f(y \mid \boldsymbol{x})$.

In fact, the assignment (2) divides the feature space into c regions, $\mathcal{R}_y = \{\boldsymbol{x} : f(y \mid \boldsymbol{x}) = \max_z f(z \mid \boldsymbol{x})\}$, $y = 0, \ldots, c-1$.

For the indicator loss, the smallest possible expected loss (i.e., the irreducible risk $\ell(g^*)$) is equal to $\mathbb{P}[Y \neq g^*(\boldsymbol{X})]$. This is called the Bayes error rate.

Pre-classifier

For a given training set τ, a classifier is often derived from a pre-classifier g_τ, which is a prediction function (learner) that can take any real value, rather than only values in the set of class labels.

A typical situation is the case of binary classification with labels -1 and 1, where the prediction function g_τ is a function taking values in the interval $[-1, 1]$ and the actual classifier is given by $\text{sign}(g_\tau)$.

It will be clear from the context whether a prediction function g_τ should be interpreted as a classifier or pre-classifier.

Classification Methods

There are many ways to fit a classifier to a training set $\tau = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$.

In subsequent lectures we will consider:

- Bayesian classification
- Linear and quadratic discriminant analysis
- Logistic classification
- K-nearest neighbors method
- Support vector machines
- Classification trees
- Neural networks

Training and Test Sets

The effectiveness of a classifier g is, theoretically, measured in terms of the risk (1), which depends on the loss function used.

Fitting a classifier to iid training data $\tau = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is established by minimizing the training loss

$$\ell_\tau(g) = \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, g(\boldsymbol{x}_i)) \tag{3}$$

over some class of functions $\mathcal{G}$.

As the training loss is often a poor estimator of the risk, the risk is usually estimated as in (3), using instead a test set $\tau' = \{(\boldsymbol{x}'_i, y'_i)\}_{i=1}^{n'}\}$ that is independent of the training set.

Loss and Confusion Matrices

Consider a classification problem with classifier g, loss function Loss, and classes $0, \ldots, c-1$.

If an input feature vector $\boldsymbol{x}$ is classified as $\widehat{y} = g(\boldsymbol{x})$ when the observed class is y, the loss incurred is, by definition, $\text{Loss}(y, \widehat{y})$.

Consequently, we may identify the loss function with a loss matrix $\mathbf{L} = [\text{Loss}(j,k), j,k \in \{0,\ldots,c-1\}]$. For the indicator loss function, the matrix $\mathbf{L}$ has 0s on the diagonal and 1s everywhere else.

Another useful matrix is the confusion matrix, denoted by $\mathbf{M}$, where the (j,k)-th element of $\mathbf{M}$ counts the number of times that, for the training or test data, the actual (observed) class is j whereas the predicted class is k.

The training/test loss of the classifier in terms of $\mathbf{L}$ and $\mathbf{M}$ is:

$$\frac{1}{n}\sum_{j,k}[\mathbf{L}\odot\mathbf{M}]_{jk}, \tag{4}$$

where $\mathbf{L}\odot\mathbf{M}$ is the elementwise product of $\mathbf{L}$ and $\mathbf{M}$.

For the indicator loss this is the misclassification error $1-\mathrm{tr}(\mathbf{M})/n$.

The expression (4) makes it clear that both the counts and the loss are important in determining the performance of a classifier.

Table: Confusion matrix for three classes.

	Predicted		
Actual	Dog	Cat	Possum
Dog	30	2	6
Cat	8	22	15
Possum	7	4	41

True/False Positive/Negative

In the spirit of hypothesis testing, it is sometimes useful to divide the elements of a confusion matrix into four groups.

- The diagonal elements are the true positive counts; that is, the numbers of correct classifications for each class. The true positive counts for the Dog, Cat, and Possum classes in Table 1 are 30, 22, and 41, respectively.
- Similarly, the true negative count for a class is the sum of all matrix elements that do not belong to the row or the column of this particular class. For the Dog class it is $22 + 15 + 4 + 41 = 82$.
- The false positive count for a class is the sum of the corresponding column elements without the diagonal element. For the Dog class it is $8 + 7 = 15$.
- Finally, the false negative count for a specific class, can be calculated by summing over the corresponding row elements (again, without counting the diagonal element). For the Dog class it is $2 + 6 = 8$.

True/False Positive/Negative

In terms of the elements of the confusion matrix, we have the following counts for class $j = 0, \ldots, c-1$:

$$\begin{aligned}
\text{True positive} \qquad & \mathrm{tp}_j = \mathbf{M}_{jj}, \\
\text{False positive} \qquad & \mathrm{fp}_j = \sum_{k \neq j} \mathbf{M}_{kj}, && \text{(column sum)} \\
\text{False negative} \qquad & \mathrm{fn}_j = \sum_{k \neq j} \mathbf{M}_{jk}, && \text{(row sum)} \\
\text{True negative} \qquad & \mathrm{tn}_j = n - \mathrm{fn}_j - \mathrm{fp}_j - \mathrm{tp}_j.
\end{aligned}$$

Misclassification Error and Accuracy

In the binary classification case ($c = 2$), and using the indicator loss function, the misclassification error (4) can be written as

$$\text{error}_j = \frac{\text{fp}_j + \text{fn}_j}{n}.$$

This does not depend on which of the two classes is considered, as $\text{fp}_0 + \text{fn}_0 = \text{fp}_1 + \text{fn}_1$.

Similarly, the accuracy measures the fraction of correctly classified objects:

$$\text{accuracy}_j = 1 - \text{error}_j = \frac{\text{tp}_j + \text{tn}_j}{n}.$$

Other Classification Metrics

In some cases, classification error (or accuracy) alone is not sufficient to adequately describe the effectiveness of a classifier.

As an example, consider the following two classification problems based on a fingerprint detection system:

1. Identification of authorized personnel in a top-secret military facility.
2. Identification to get an online discount for some retail chain.

Both problems are binary classification problems. However, a false positive in the first problem is extremely dangerous, while a false positive in the second problem will make a customer happy.

Other Classification Metrics

Suppose the classifier in the top-secret facility has the following confusion matrix.

	Predicted	
Actual	authorized	non-authorized
authorized	100	400
non-authorized	50	100,000

The accuracy of classification is equal to

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} = \frac{100 + 100,000}{100 + 100,000 + 50 + 400} \approx 99.55\%.$$

However, we can see that in this particular case, accuracy is a problematic metric, since the algorithm allowed 50 non-authorized personnel to enter the facility.

Other Classification Metrics

One way to deal with this issue is to modify the loss function to give a much higher loss to non-authorized access.

Thus, instead of an (indicator) loss matrix $\mathbf{L} = \mathbf{I}$ (identity matrix), we could for example take the loss matrix

$$\mathbf{L} = \begin{pmatrix} 0 & 1 \\ 1000 & 0 \end{pmatrix}.$$

An alternative approach is to keep the indicator loss function and consider additional classification metrics.

Next, we give a list of commonly used metrics, calling an object whose actual class is j a "j-object".

- The precision (also called *positive predictive value*) is the fraction of all objects classified as j that are actually j-objects:

$$\text{precision}_j = \frac{\text{tp}_j}{\text{tp}_j + \text{fp}_j}.$$

- The recall (also called *sensitivity*) is the fraction of all j-objects that are correctly classified as such:

$$\text{recall}_j = \frac{\text{tp}_j}{\text{tp}_j + \text{fn}_j}.$$

- The specificity measures the fraction of all non-j-objects that are correctly classified as such:

$$\text{specificity}_j = \frac{\text{tn}_j}{\text{fp}_j + \text{tn}_j}.$$

- The F_β score is a combination of the precision and the recall and is used as a single measurement for a classifier's performance:

$$F_{\beta,j} = \frac{(\beta^2 + 1)\,\text{tp}_j}{(\beta^2 + 1)\,\text{tp}_j + \beta^2\,\text{fn}_j + \text{fp}_j}.$$

Example: Comparing Classifiers

Suppose for the classification of authorized personnel in a top-secret military facility, we have another classifier, with confusion matrix

Table: Confusion matrix for authorized personnel classification, using a different classifier (Classifier 2).

	Predicted	
Actual	Authorized	Non-Authorized
authorized	50	10
non-authorized	450	100,040

Example: Comparing Classifiers

Various metrics for these two classifiers are show in the table below.

Metric	Classifier 1	Classifier 2
accuracy	9.955×10^{-1}	9.954×10^{-1}
precision	$\mathbf{6.667 \times 10^{-1}}$	$\mathbf{1.000 \times 10^{-1}}$
recall	2.000×10^{-1}	8.333×10^{-1}
specificity	9.995×10^{-1}	9.955×10^{-1}
F_1	3.077×10^{-1}	1.786×10^{-1}

In this case we prefer Classifier 1, which has a much higher precision.

Multilabel Classification

In standard classification the classes are assumed to be mutually exclusive. For example a satellite image could be classified as "cloudy", "clear", or "foggy".

In multilabel classification the classes (often called labels) do not have to be mutually exclusive. In this case the response is a subset $\mathcal{Y}$ of some collection of labels $\{0, \ldots, c-1\}$.

Equivalently, the response can be viewed as a binary vector of length c, where the y-th element is 1 if the response belongs to label y and 0 otherwise.

In the satellite image example, add two labels, such as "road" and "river" to the previous three labels. Clearly, an image can contain both a road and a river. In addition, the image can be clear, cloudy, or foggy.

Hierarchical Classification

In hierarchical classification a hierarchical relation between classes/labels is taken into account during the classification process.

Usually, the relations are modeled via a tree or a directed acyclic graph.

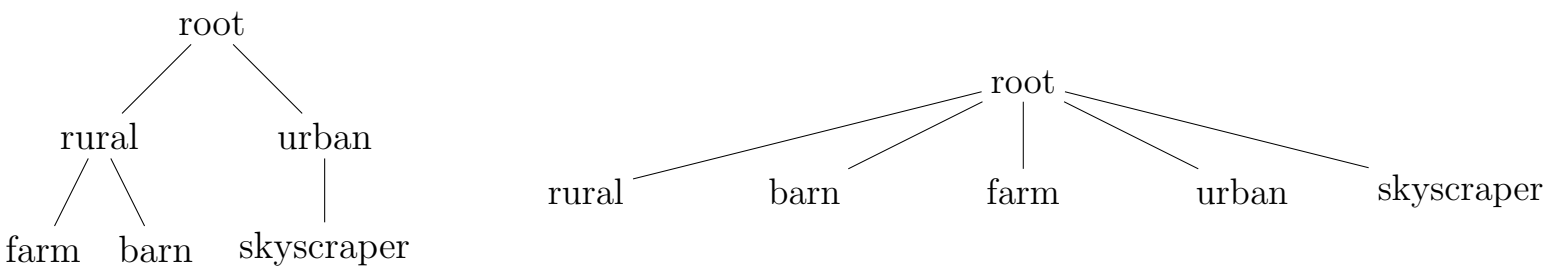


Figure: Hierarchical (left) and non-hierarchical (right) classification schemes. Barns and farms are common in rural areas, while skyscrapers are generally located in cities. While this relation can be clearly observed in the hierarchical model scheme, the connection is missing in the non-hierarchical design.

Metrics for Multilabel Classification

In multilabel classification, both the prediction $\widehat{\mathcal{Y}} := g(\boldsymbol{x})$ and the true response $\mathcal{Y}$ are subsets of the label set $\{0, \ldots, c-1\}$. A reasonable metric is the so-called exact match ratio, defined as

$$\text{exact match ratio} = \frac{\sum_{i=1}^{n} \mathbb{I}\{\widehat{\mathcal{Y}}_i = \mathcal{Y}_i\}}{n}.$$

The exact match ratio is rather stringent, as it requires a full match. The following metrics could be used instead.

- The *accuracy* is defined as the ratio of correctly predicted labels and the total number of predicted and actual labels:

$$\text{accuracy} = \frac{\sum_{i=1}^{n} |\mathcal{Y}_i \cap \widehat{\mathcal{Y}}_i|}{\sum_{i=1}^{n} |\mathcal{Y}_i \cup \widehat{\mathcal{Y}}_i|}.$$

Metrics for Multilabel Classification

- The *precision* is defined as the ratio of correctly predicted labels and the total number of predicted labels:

$$\text{precision} = \frac{\sum_{i=1}^{n} |\mathcal{Y}_i \cap \widehat{\mathcal{Y}}_i|}{\sum_{i=1}^{n} |\widehat{\mathcal{Y}}_i|}.$$

- The *recall* is defined as the ratio of correctly predicted labels and the total number of actual labels:

$$\text{recall} = \frac{\sum_{i=1}^{n} |\mathcal{Y}_i \cap \widehat{\mathcal{Y}}_i|}{\sum_{i=1}^{n} |\mathcal{Y}_i|}.$$

- The *Hamming loss* counts the average number of incorrect predictions for all classes:

$$\text{Hamming} = \frac{1}{n\,c} \sum_{i=1}^{n} \sum_{y=0}^{c-1} \mathbb{I}\{y \in \widehat{\mathcal{Y}}_i\}\, \mathbb{I}\{y \notin \mathcal{Y}_i\} + \mathbb{I}\{y \notin \widehat{\mathcal{Y}}_i\}\, \mathbb{I}\{y \in \mathcal{Y}_i\}.$$